

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено

Завідувач кафедри

О.В. Коваль  
(підпис) (ініціали, прізвище)

“ ” 2019р.

ДИПЛОМНА РОБОТА

на здобуття ступеня бакалавра

з напрямку підготовки 6.050101 “Комп’ютерні науки”

на тему «Формування сцени проведення гідроакустичних експериментів в складі моделюючого комплексу з використанням ПС»

Виконав: студент 4-го курсу, групи ТМ-52

Кравчук Олексій Ігорович  
(прізвище, ім’я, по батькові)

\_\_\_\_\_  
(підпис)

Керівник асист. Швайко В.Г.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Рецензент доцент, к.т.н Баранюк О.В.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019 року  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 6.050101 “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

О.В. Коваль  
(підпис)

”    ”    \_\_\_\_\_ 2019р.

## ЗАВДАННЯ

на дипломну роботу студенту

Кравчуку Олексію Ігоровичу

(прізвище, ім’я, по батькові)

1. Тема роботи «Формування сцени проведення гідроакустичних експериментів в складі моделюючого комплексу з використанням ПС»

керівник роботи \_\_\_\_\_ асист. Швайко Валерій Григорович  
(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ”22” 03 2018р. №1004-с

2. Строк подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи мова програмування C#, платформа Microsoft Visual Studio 2010

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати існуючі програмні засоби, розробити застосунок, спроектувати структуру бази даних, спроектувати архітектуру програми, реалізувати інтерфейс користувача системи, розробити інструкцію користувача

5. Перелік ілюстративного матеріалу

Мета розробки, Актуальність, Фільтр Гауса, Алгоритм фільтрації зображення, Засоби розробки, Зображення формування сцени, Інтерфейс користувача

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання ” 1 ” грудня 2018 р.

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	01.12.2018	
2.	Вивчення та аналіз задачі	02.12.2018 – 19.12.2018	
3.	Розробка архітектури та загальної структури системи	19.12.2018 – 29.12.2018	
4.	Розробка структур окремих підсистем	09.01.2019 – 23.01.2019	
5.	Програмна реалізація системи	09.01.2019 – 19.01.2019	
6.	Оформлення пояснювальної записки	22.01.2019 – 12.03.2019	
7.	Захист програмного продукту	17.05.2019	
8.	Передзахист	31.05.2019	
9.	Захист	17.06.2019 – 21.06.2019	

Студент \_\_\_\_\_ Кравчук О.І.  
(підпис) (прізвище та ініціали,)

Керівник роботи \_\_\_\_\_ Швайко В.Г.  
(підпис) (прізвище та ініціали,)

## **АНОТАЦІЯ**

Метою роботи було створення програмного модулю для формування сцени проведення гідроакустичних експериментів в складі моделюючого комплексу з використанням ГІС. Розроблені функції формування сцени, додавання на карту гідроакустичного датчика і судна. В роботі представлені результати експерименту в умовах китайського.

Записка містить 44 сторінок, 19 рисунків

## **ABSTRACT**

The purpose of the work was to create a software module for the formation of the stage of conducting hydroacoustic experiments in the composition of the simulation complex using GIS. The functions of forming the scene, the addition of a hydroacoustic sensor and a vessel to the card are developed. The results of the experiment are presented in Chinese. The note contains

The note contains 44 pages, 19 images

## **АНОТАЦИЯ**

Метою работы было создание программного модуля для формирования сцены проведения гидроакустических экспериментов в составе моделирующего комплекса с использованием ГИС. Разработлены Функции формирования сцены, добавления на карту гидроакустического датчика и судна. В работе представлены результаты эксперимента в условиях китайского.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	6
ВСТУП.....	7
1 ПОСТАНОВКА ЗАДАЧІ .....	9
2 ЗАГАЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ МОДЕЛЮВАННЯ.....	10
2.1 Опис моделювання системи .....	11
2.2 Технології моделювання.....	<b>12</b>
2.3 Технологія Батиметрії.....	14
2.4 Архітектура моделюючого комплексу.....	16
3 ЗАСОБИ РОЗРОБКИ .....	19
3.1 Середовище розробки Visual Studio 2010.....	16
3.2 Windows Forms .....	20
3.3 Огляд функціоналу ArcMap .....	22
3.4 Огляд SQL.....	24
4 ОБЧИСЛЮВАЛЬНІ МЕТОДИ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ.....	27
5.1 Середовище розробки .....	28
5.2 Технології, використані при розробці програмного засобу .....	30
5 Опис програмної реалізації.....	32
5.1 Опис реалізації.....	32
5.2 Опис Баз Даних.....	35
5.3 Висновки до розділу.....	38
6 Робота користувача з програмою .....	38

6.1 Системні вимоги.....	38
6.2 Робота з програмою.....	38
6.3 Висновки до розділу.....	45
ВИСНОВКИ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

### Умовні скорочення:

СУБД– СИСТЕМА УПРАВЛІННЯ БАЗАМИ ДАНИХ

БД – база даних

ГІС– геоінформаційна система

ГАП – Гідроакустичні процеси



## ВСТУП

Експериментальним дослідженням гідроакустичних процесів пов'язаних з вивченням особливостей шумовипромінювання морських об'єктів кінцевою метою яких є класифікації та ідентифікації цих об'єктів на основі використання ряду відповідних алгоритмів, таких як генерації сигналів, виділення сигналів з поточного стану загального гідроакустичного поля акваторії, класифікації та кінцевої ідентифікації об'єктів за конкретними ознаками і властивостями у сьогоdnішній час приділяється значна увага.

За звичай ці дослідження виконуються на основі натурних експериментів, які проводяться в різних акваторіях з різними об'єктами та підтверджують високу ефективність дослідження спектрально-енергетичних характеристик виділених сигналів шумоізлучаючих морських об'єктів, при цьому важлива процедура самої оцінки ефективності ідентифікації, класифікації та ймовірності правильного виявлення, класифікації, пропуску помилкових тривог.

Експериментально визначені характеристики шумовипромінювання морських об'єктів дозволяють розробити різнопланові морські охоронні системи. Ці системи забезпечують визначення, ідентифікацію, пеленгування, визначення дальності, швидкості, курсових кутів і побудова трас руху морських об'єктів.

Роботи, пов'язані з проведенням кожного окремого натурального експерименту є ресурс витратними та вимагають додаткової виконання процедур: фіксації умов експерименту, тимчасової прив'язки подій і ретельного їх опису для зберігання і подальшого поглибленого аналізу і обробки.

Такі процедури передбачають наявність засобів ідентифікації умов проведення експерименту, наприклад, район проведення, параметри акваторії, набір приладів виявлення сигналів (їх розташування, характеристики). Перераховані процедури можна умовно звести до поняття створення і опису сцени моделювання.

Крім цього експерименти передбачають різноманітні методики фіксації сигналів, різноманітність їх спектрально-енергетичних характеристик, що вимагає часової



прив'язки ходу проведення експерименту та використання ретельного опису послідовності проведення окремих кроків експерименту, що можна умовно звести до процедури організації сценарію проведення експерименту.

## 1. ПОСТАНОВКА ЗАДАЧІ

Метою роботи є розробка модулю для проведення гідроакустичних експериментів з використанням цифрових моделей рельєфу, цифрової картографічної інформації і оперативної і прогностичної інформації про режим водних об'єктів та подальше інтегрування модулю у ГІС систему. Призначенням даного програмного засобу є надання чіткого зрозумілого інтерфейсу для формування сцени проведення гідроакустичних експериментів, які пропонує обрана ГІС система ArcGis Desktop.

При виконанні роботи потрібно провести моделювання сцени гідроакустичних досліджень. Інформаційною базою для моделювання повинні стати такі джерела: база даних для гідроакустичного датчика; база даних кораблів; карта китайського моря.

## 2. ЗАГАЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ МОДЕЛЮВАННЯ

На даний момент не існує універсальної системи моделювання гідроакустичних процесів, що показують необхідність подальшої розробки технологій та методів для таких досліджень

В роботі пропонується побудувати спеціалізований програмно комплекс моделювання гідроакустичних процесів з використанням ГІС. Такі підходи дозволять значно поширити коло задач дослідження та вирішити проблеми пов'язані з багатоманітністю методик фіксації сигналів. Крім цього використання сценарного підходу при проведенні досліджень забезпечить побудову надійної системи збереження параметрів експериментів та можливості формування правил оцінювання результатів моделювання.

Загальною метою побудови моделюючого комплексу є вивчення та формування оцінок різних комп'ютерних моделей гідроакустичних процесів з позиції вірності відображення реальних фізичних процесів, поширення акустичних сигналів, а також опрацювання та удосконалення алгоритмів їх розпізнавання і на цій основі проведення класифікацій та ідентифікацій випромінюючих об'єктів у районах спостереження.

Програмний комплекс створений вирішувати задачі моделювання:

1. формування сцени для гідроакустичних досліджень;
2. моделювання об'єктів;
3. моделювання гідроакустичного датчика на поверхні моря;

Крім того, комплекс повинен вирішувати завдання організації дослідного процесу, що полягає в проведенні ряду експериментів:

- облік експериментів,
- організацію сцен,
- формування сценаріїв,
- оцінку випробуваних в системі моделей.

В цьому випадку комплекс вирішує основні завдання управління роботою на основі рішення задач :

1. включення процедур управління сценами моделювання, що дозволяє виконати: документування властивостей сцени (реєстрація, модифікація і видалення), формування з використанням засобів типової геоінформаційної системи наповнення сцени , шумовипромінюючих об'єктів і гідроакустичними приладами фіксації шумів;
2. використання механізмів управління сценаріями (послідовність кроків моделювання) при виконанні експерименту, які виконують процедури документування сценаріїв, покрокове або комплексне виконання включених в сценарій окремих етапів моделювання з фіксацією оцінок їх виконання в програмі;
3. ведення бібліотеки моделей з можливостями додавання, модифікації, видалення та активізації окремих математичних моделей;
4. включення механізмів дослідження моделі окремо поза сценарієм;
5. формування та ведення бази даних з описом експериментів і довідником системи БД;
6. включення засобів оцінювання результатів експерименту на основі використаних процедур формування та використання бази знань;
7. збереження кінцевих результатів в базу даних.

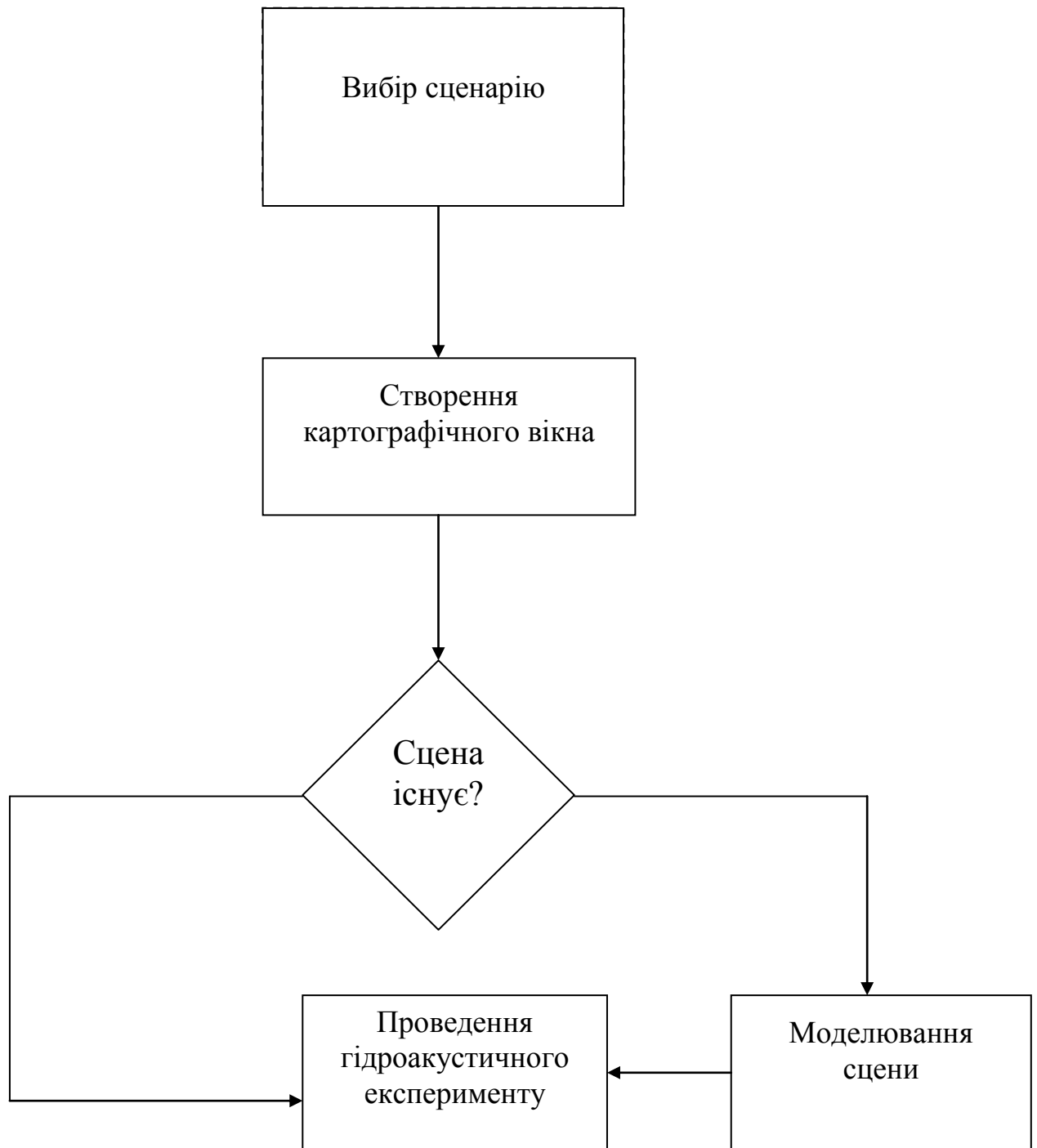
## **2.1 Опис моделювання системи**

Комплекс досліджень являє собою деяку систему проведення модельних гідроакустичних експериментів з використанням елементів сценарного підходу.

Сутність сценарного підходу є створення сценаріїв аналізу будь-яких процесів моделювання, тобто деяка фіксована послідовності примінення інструментів моделювання та процедур оцінювання якості обраних моделей із

завданням параметрів, які визначають їх використання. Система повинна мати засоби, що дозволяють фіксувати сценарії, забезпечувати їх виконання, а також сформувати поетапну оцінку як роздільної здатності методів виконати покладені на них завдання, так і оцінку результатів отриманих за їх допомогою.

Загальна схема проведення експериментів використовуючи сценарний підхід (Рисунок 2.1).



Рисинук 2.1 – схема моделювання експерименту

## 2.2 Технології моделювання

Напрямок досліджень за допомогою натурних експериментів, дозволяє виробляти загальні підходи у організації навігаційних систем, створенні систем акустичної томографії, досліджувати якісні характеристики систем виявлення та визначення морських об'єктів у акваторіях.

В роботі надані результати з розробки геоінформаційної системи (ГІС) призначеною для моделювання та дослідження гідроакустичних полів.

Основними елементами ГІС є:

- ГІС-інтерфейс. Служить для взаємодій користувача з системою і інтеграції складових частин ГІС для забезпечення функціональності цілої системи;
- База знань. Включає онтологію, яка описує сутність предметної області і відносини між ними. Другою частиною бази знань є цілісність об'єктів (сервер-об'єктів), що представляють реальні сутності предметної області;
- Сервер картографічної інформації. Призначений для візуалізації картографічних даних, роботи з картографічними даними в єдиному координатному просторі;
- Сервіс електронного документообігу. Забезпечує формування звітів і обмін звітами між віддаленими користувачами;
- Комплекс розрахункових моделей. Призначений для прогнозування і апроксимації параметрів середовища, реалізації алгоритмів розрахунку акустичного поля, апроксимації вихідних даних моделювання, а також реалізації інших розрахункових алгоритмів.

Обмеження системи є не визначеність понять сцен та сценаріїв, фіксований перелік моделей розрахунків, відсутність процедур ідентифікації об'єктів

Таким чином, для вирішення задач управління гідроакустичними експериментам з метою аналізу та оцінки вирішальної здатності алгоритмів генерації та аналізу гідроакустичних сигналів подальшому проведенні процедур класифікації та ідентифікації морських штучних шумовипромінюючих об'єктів потрібна окрема система моделювання.

## 2.3 Технологія Батиметрії

Для зондування дна водойм (батиметрія) лазерним випромінюванням використовується лазер видимого діапазону, синього або зеленого кольору. Мінімум поглинання дистильованої води знаходиться в синій частині спектра, проте в реальних умовах через розчинених у воді речовин мінімум зміщений в зелену область спектра.

Тому неодимові імпульсні лазери з подвоєнням частоти найкращим чином підходять для профілювання дна: загасання світла з довжиною хвилі 532 нм в прісній і морській воді мінімально. При проведенні авіаційної батиметричної зйомки виходить карта, на якій різні кольори відповідають різним глибинам (рисунк 2.2).

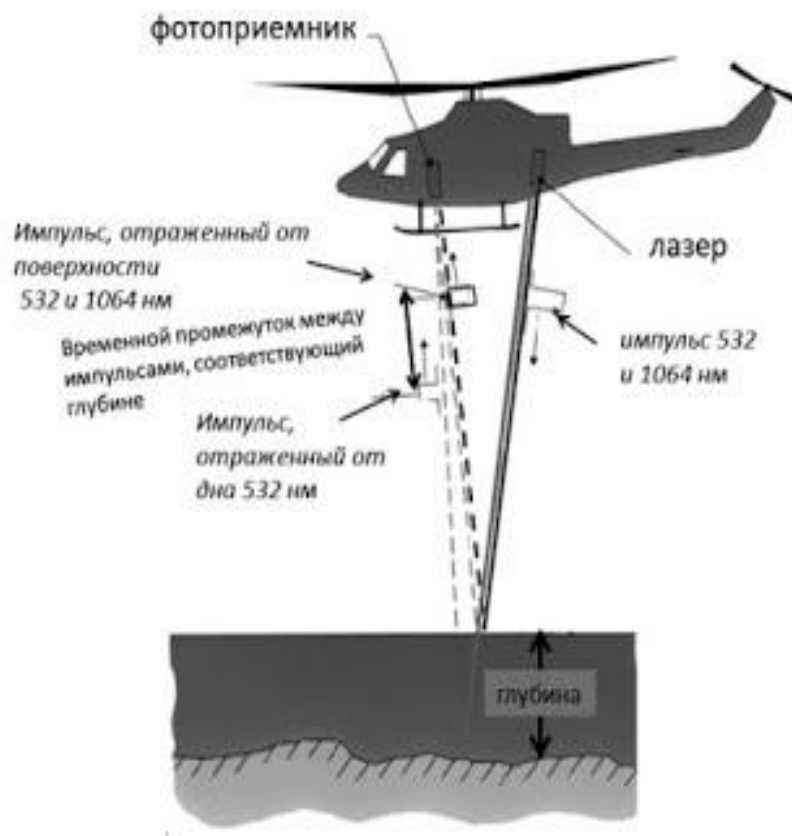


Рисунок 2.2 – Приклад Батиметрії

Для вимірювання профілю дна використовується зазвичай імпульсний неодимовий лазер з подвоєнням частоти. Пройшовши через нелінійний кристал,

частина випромінювання з довжиною хвилі 1064 нм перетворюється в світло 532 нм. Лазерний передавач посилає по одній і тій же траєкторії імпульс з двома довжинами хвиль, видимого та ближнього інфрачервоного діапазонів.

Оскільки поглинання інфрачервоного випромінювання в воді досить сильно, то сигнал на довжині хвилі 1064 нм не доходить до дна і може бути отриманий тільки від поверхні води. З іншого боку, зелений промінь глибше проникає через товщу води і відбивається від дна.

У Фотоприймальному каналі 532 нм спостерігаються послідовно два імпульси, відповідні відображенням від поверхні води і від дна водойми. Імпульс лазера відбивається як від поверхні води, так і від дна водойми. На фотопримачі спостерігаються послідовно два піки, інтервал між якими відповідає глибині.

В основному, батиметрія оперує даними, одержуваними за допомогою гідроакустичних лотів (ехолотів, або сонарів), вмонтованих в дно судна. В реальних умовах лазерне сканування не завжди може скласти конкуренцію звуковий ехолокації: максимальна глибина, на якій вдається зареєструвати сигнал, відбитий від дна, для найчистішої морської води не перевищує 70 м. У річках, в каламутній воді глибина сканування обмежена всього декількома метрами.

Однак в деяких випадках лазерна батиметрія має незаперечні переваги. За допомогою лазерного сканера можна проводити профілювання морського дна з літака або вертольота з великою швидкістю на дуже великих територіях. Акустичне зондування дуже вразливе в місцях, багатих на обмілини і тому більш придатна для використання в глибоких водах.

Гідроакустики забезпечує радіус огляду приблизно в два рази більший, ніж глибина. Це означає, що охоплення рельєфу дна обмежений на мілководді і в прибережних зонах. З водного транспорту гідроакустичному методами важко сканувати перемежовуються мілини і мілководді.

Повітряна лазерна батиметрія, навпаки, забезпечує можливість великого охоплення ділянок рельєфу дна на мілководді, недискретні, цілісний потік даних через розділ суша-море. У реальній системі дзеркальне відображення лазерного



променя від поверхні води при вітряну погоду, яка призводить до утворення дрібних хвиль, істотно ускладнюється, оскільки ці дрібні хвилі діють як грані дзеркала.

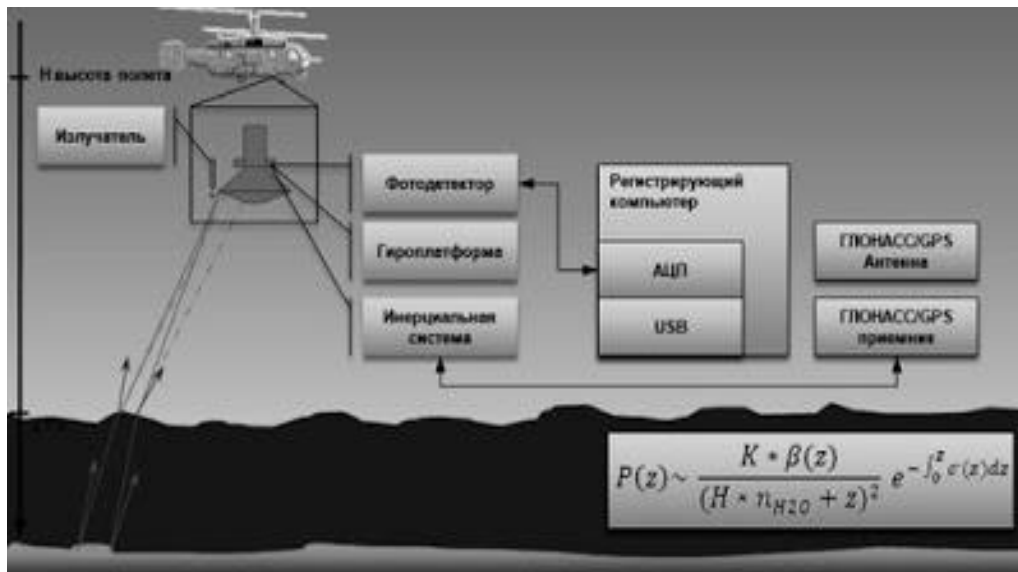


Рисунок 2.3– Батиметрія

У спокійних умовах поверхня являє собою одне велике дзеркало і, таким чином, спрямовує відбите випромінювання прямо на приймач. Слід зазначити, що вихідний лазерний промінь при скануванні утворює кут до  $20^\circ$  по відношенню до вертикалі.

Зелений лазерний промінь на 532 нм при проходженні морської поверхні заломлюється на кут близько  $15^\circ$  в залежності від структури поверхневих хвиль. Тому датчик також повинен враховувати коефіцієнт заломлення при обчисленні глибини дна або об'єкта

## 2.4 Архітектура моделюючого комплексу

Узагальнена архітектура моделюючого ко

мплексудля ідентифікації, класифікації та визначення параметрів руху морських об'єктів. Модель представлена наступними структурними компонентами (Рисунок. 2.1):

1. Моделювання морського середовища - виконує функції задання середовища моделювання (шум моря, параметри дна, берегова лінія, температури води,

глибина занурення об'єкта і таке інше), параметрів морських об'єктів (тип, розмір, напрямок, швидкість руху);

2. Моделювання гідроакустичної системи - задає конфігурацію фіксуючих приладів та їх параметри (тип, розміри, чутливість)
3. Компонент «Аналіз гідроакустичних сигналів» являє собою інструментальні методи та засоби аналізу сигналів (БПФ, цифрова фільтрація, спектральний, частотний, кореляційний аналіз та інші);
4. Компонент «Розпізнавання морських об'єктів» містить набір інструментальних засобів для визначення об'єктів, класифікації та визначення параметрів руху об'єктів, включаючи методи нечіткої логіки;
5. Компонент «Введення бази знань предметної області» виконує функції введення накопиченого досвіду, створення шаблонів, онтологій, метаописів морських об'єктів, кластеризації знань;
6. Компонент «Машина виведення» призначена для організації логічного висновку на основі накопичених знань, включаючи засоби композиції продукційних правил, самонавчання та адаптації;
7. Інформаційне сховище є сполучною ланкою між структурними компонентами моделюючої системи і містить як безпосередньо дані для опису предметної області, так і посилання на Web-сервіси обробки, аналізу, ідентифікації та класифікації даних предметної області;

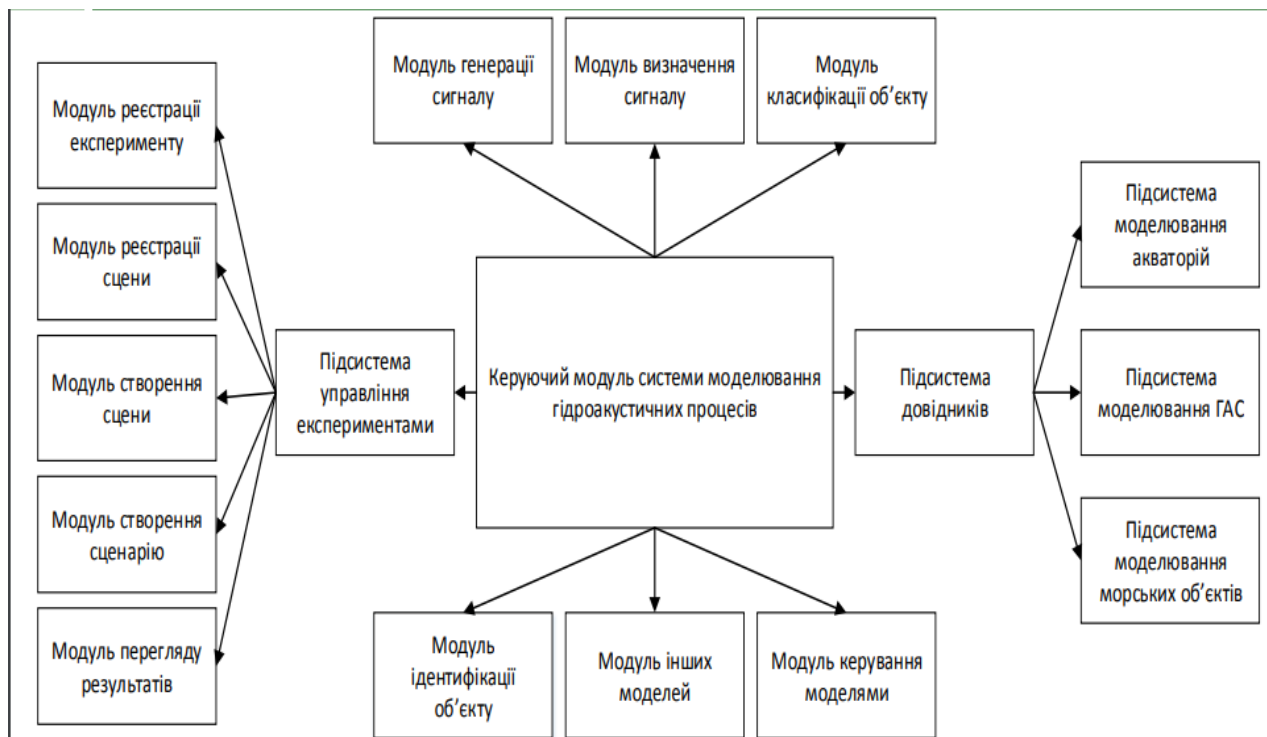


Рисунок 2.4- схема керуючого модулю

## 3 ЗАСОБИ РОЗРОБКИ

Важливим чинником, під час розробки програмного продукту, є вибір засобів програмної реалізації технологій. Середовищем розробки модуля було обрано Microsoft Visual Studio 2010. Для створення графічного інтерфейсу модулю використовувався інтерфейс додатків Windows Forms, що є частиною Microsoft .NET Framework.

### 3.1 Середовище розробки Visual Studio 2010

Середовище розробки Visual Studio 2010(рисунок 3.1) дозволяє швидко і якісно писати код. За допомогою Visual Studio 2010 можливо легко переглянути структури викликів та пов'язані функції, повернені значенні та стан тестування. Також можна перевірити код, знайти помилки та отримати варіанти їх усунення.

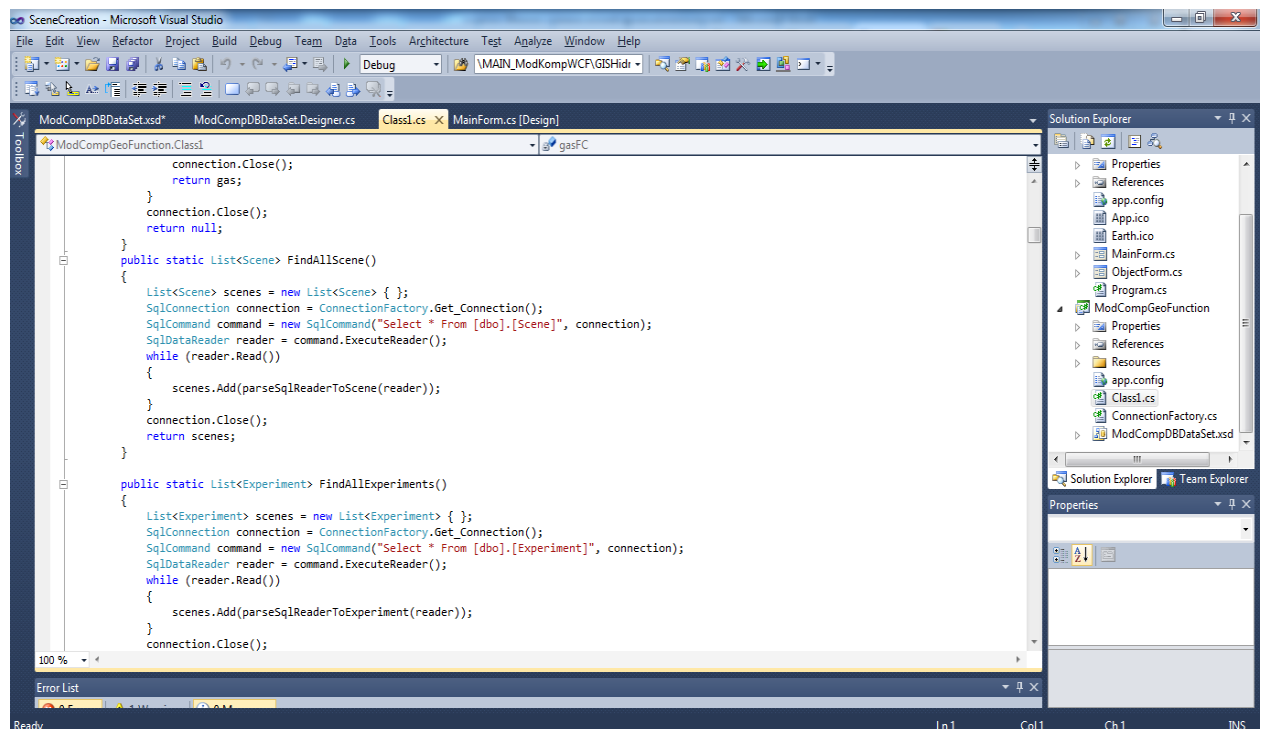


Рисунок 3.1 – Вигляд Visual Studio 2010

З кожним роком інструментарій Microsoft Visual Studio оновлюється і постійно розширюється, а компанія Microsoft постійно намагається покращити зручність середовища розробки, з урахуванням побажань розробників з усього світу,

які користуються Microsoft Visual Studio. В результаті понад 30 мільйонів завантажень на сьогоднішній день, що робить даний продукт одним із найпопулярніших для розробників додатків.

Кожен з розробників може завантажити собі 60-денну безкоштовну пробну версію Visual Studio і повністю ознайомитись з повним функціоналом середовища розробки, протестувати його на власному досвіді і зробити для себе певні висновки. А потім, в залежності від потреб, залишитися на версії Community, якщо інструментарій використовується лише для освітніх цілей, або перейти на платну версію в тому разі, коли розроблений додаток буде використовуватись як комерційний продукт.

Варто виділити модульний підхід до процесу установки інструментів. Серед великої кількості різних модулів, що дозволяють програмісті розробляти додатки різного рівня і для різних платформ, тепер можна обрати лише ті, що потрібні конкретно для реалізації вашої задачі. Це прискорює процес установки і не загромаджує ваш диск непотрібними інструментами, а отже і загальний час, що витрачається на розробку програмного продукту загалом.

Ще одним важливим нововведенням для сімейства Microsoft Visual Studio стала інтеграція з хмарними сервісами платформи Azure. Ця інтеграція дозволяє полегшити налаштування так створення програмних рішень в хмарі Azure напряду із середовища IDE.

### **3.2 Windows Forms**

Windows Form дозволяє розробляти інтелектуальні клієнти. Інтелектуальний клієнт - це програма з повнофункціональним графічним інтерфейсом, просте в використанні, здатне працювати при наявності або відсутності підключення до Інтернету і використовує більш безпечний доступ до ресурсів на локальному комп'ютері в порівнянні з традиційними додатками Windows.

Windows Forms - це технологія інтелектуальних клієнтів для платформи .NET Framework, набір керованих бібліотек, що спрощують виконання поширених

завдань, таких як читання і запис в файлової системі. При використанні середовища розробки, як Microsoft Visual Studio, можна створювати додатки інтелектуальних клієнтів Windows Forms, які відображають інформацію, запитують введення від користувачів і взаємодіяти з віддаленими комп'ютерами по мережі.

У Windows Form форма - це візуальна поверхня, на яку виводиться інформація для користувача. Зазвичай додаток Windows Forms будується шляхом додавання елементів управління на форму і написання коду для реагування на дії користувача, такі як клацання миші або натискання клавіш. Елемент управління - це окремий елемент призначеного для користувача інтерфейсу, використаний для відображення або введення даних.

Windows Forms(рисунок3.2) включає широкий вибір елементів управління, які можна додавати на форми: текстові поля, кнопки, списки, що розкриваються, перемикачі та навіть веб-сторінки. Список всіх елементів керування, які можна використовувати в формі, представлені в розділі Елементи управління для використання в формах Windows Forms. Якщо існуючий елемент керування не задовільняє потребам, в Windows Forms можна створювати власні елементи управління за допомогою класу UserControl.

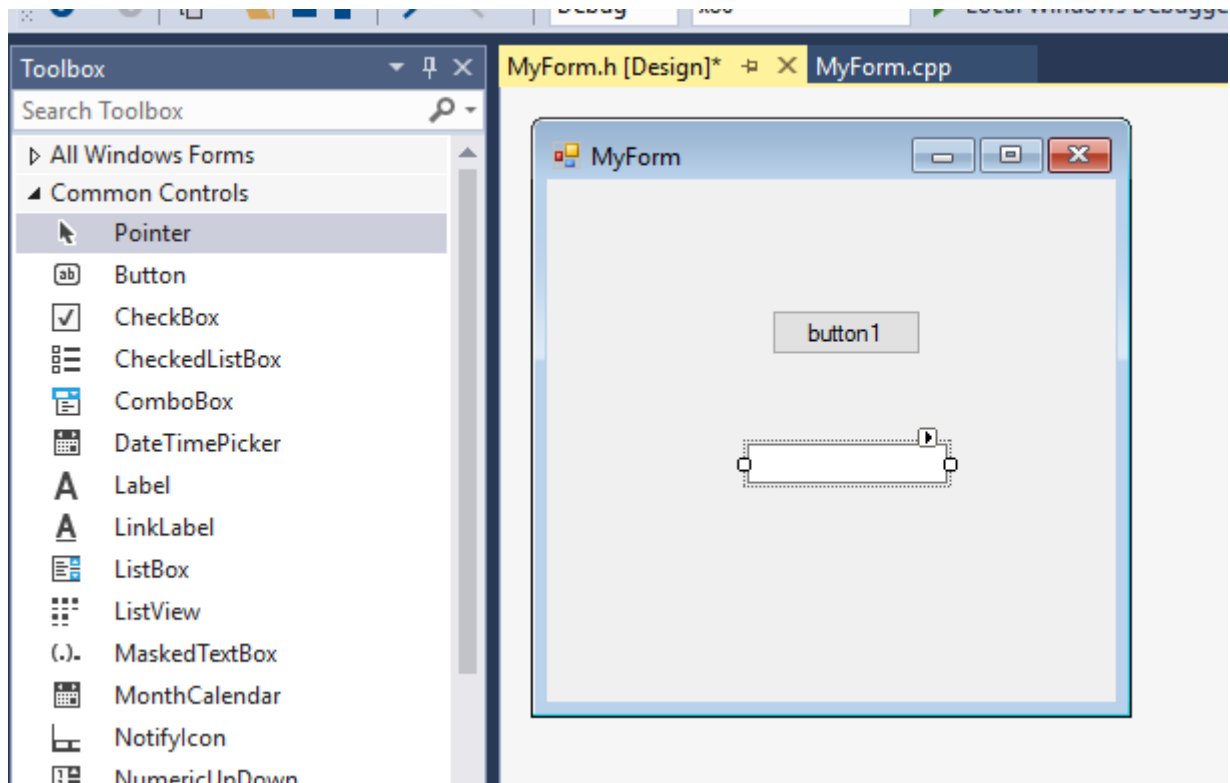


Рисунок 3.2 – Приклад конструктора Windows Forms

До складу Windows Forms входять багатофункціональні елементи призначені для користувача, що дозволяють відтворювати можливості таких складних додатків, як Microsoft Office. Використовуючи елементи керування ToolStrip і MenuStrip, можна створювати панелі інструментів і меню, що містять тексти і малюнки, підменю та інші елементи управління, такі як текстові поля, таблиці і поля зі списками.

### **3.3 Огляд функціоналу ArcMap**

ArcGIS - це програмне забезпечення для побудови ГІС будь-якого рівня. ArcGIS допомагає використовувати географічну інформацію для проведення аналізу, кращого розуміння даних і прийняття більш інформованих рішень.

Завданням дипломної роботи є розробка модулю для інтеграції в систему ArcMap, яка є основною прикладною програмою ArcGIS, яка використовується для вирішення різних GIS-задач, як загального профілю, так і вузьконаправлених. Тому дуже важливо мати навички роботи з базовими функціями програми. Типові задачі, що вирішуються засобами ArcMap:

- Робота з картами - можливість відкрити документи ArcMap і працювати з ними, щоб вивчати інформацію, розглядати карти, вимикати і вмикати шари, створювати запити до атрибутів даних, представлених на карті, візуалізувати географічну інформацію з карти.
- Друк карт - за допомогою ArcMap можна роздруковувати карти різного рівня складності та розмірів.
- Компіляція і редагування наборів геоданих - ArcMap пропонує основну можливість автоматизації робіт з наборами бази даних. ArcMap підтримує повне редагування користувацьких налаштувань. Можна вибрати шари для редагування в документі карти, нові або оновлені об'єкти зберігаються в наборі даних шару.
- Використання геообробки для автоматизації роботи і виконання аналізу - ГІС

використовується не тільки для візуалізації, але і для аналізу. ArcMap дає можливість запуску моделей, а також перегляду і роботи з результатами роботи у вигляді карти. Геообробку можна використати для аналізу, а також для автоматизації багатьох типових задач, наприклад, створення багат шарових карт, відновлення пошкоджених посилань на дані в наборі документів карти, виконання різних операцій над геоданими.

- Організація баз геоданих, документів ArcGIS і управління ними - У ArcMap є вікно Catalog, яке дозволяє організувати всі набори ГІС-даних і бази даних, документи карти та інші ArcGIS файли, інструменти геообробки і безліч інших елементів ГІС. Можна також налаштувати і організувати схему бази даних у вікні Каталогу.
- Публікація документів карт в якості картографічних сервісів за допомогою ArcGIS for Server - вміст ArcGIS можна помістити в веб-середовище шляхом картинки географічної інформації у вигляді серії картографічних сервісів. ArcMap надає можливість звичайному користувачеві публікувати документи карт в якості картографічного сервісу.
- Спільна робота з картами, шарами, моделями обробки і базами даних з іншими користувачами - ArcMap містить інструменти, що спрощують процеси упаковки і спільної роботи над наборами геоданих з іншими користувачами. Крім того, ви можете помістити ваші карти і дані в загальному доступі за допомогою ArcGIS Online.

Переваги використання ArcMap:

- Дані збираються швидко, з великою точністю
- Дані поверхні мають високу щільність. Висока щільність точок також покращує результати різних досліджень, наприклад, вивчення меж області затоплення
- Збір даних висот під щільним пологом лісу, де фотограмметричних методом не досягається достатня точність поверхні через щільність лісового пологу
- Лідар використовує активний світловий сенсор і може збирати дані в будь-який час доби, на відміну від традиційних фотограмметричних методів.



- Лідар не дає геометричних спотворень, як наприклад, радіолокаційна станція бокового огляду.
- Лідарні дані можуть інтегруватися з іншими джерелами даних.

### 3.4 Огляд SQL

**SQL** (англ. *Structured query language* — мова структурованих запитів) — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних. Сама по собі SQL не є ані системою керування базами даних, ані окремим програмним продуктом. На відміну від дійсних мов програмування (C або Pascal), SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати як інструкції для керування даними. Окрім цього, стандарт SQL містить функції для визначення зміни, перевірки та захисту даних.

SQL — це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також керування базами даних. Багато баз даних підтримує SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення і вилучення даних за допомогою використання системи керування і адміністративних функцій. SQL також включає CLI (Call Level Interface) для доступу і керування базами даних дистанційно.

Перша версія SQL була розроблена на початку 1970-х років у IBM. Ця версія мала назву **SEQUEL** і була призначена для обробки та пошуку даних, що містилися в реляційній базі даних IBM, System R. Мова SQL надалі була стандартизована Американськими Держстандартами (ANSI) в 1986. На початку SQL була запланована як мова запитів і управління даними, а пізніші модифікації SQL створені продавцями системи управління базами даних, які додали процедурні конструкції, control-of-flow команд і темпоральні розширення мов. З випуском

стандарту SQL:1999 такі розширення були формально запозичені як частина мови SQL через Persistent Stored Modules (SQL/PSM).

Критика SQL включає відсутність крос-платформенності, невідповідну обробку відсутніх даних (дивіться Null (SQL)). Часто це неоднозначна граматики і семантика мови.

Мова SQL поділяється на кілька видів елементів:

- Пункти, що є складовими частинами запитів.
- Вирази, які можуть генерувати скалярні значення, або таблиці з стовпчиками і рядками даних
- Предикати (англ. Predicates), які описують умови, результатом яких є значення тризначної логіки SQL (true/false/unknown) або Булеві значення істинності і які використовуються для обмеження ефекту інструкцій та запитів, або для зміни потоку виконання програми.
- Запити (англ. Queries), які отримують дані на основі заданих критеріїв.
- Інструкції (англ. Statements), які чинять дію на схему даних чи самі дані, або контролюють транзакції, потік виконання програми, з'єднання, сесії, та виконують діагностику.
- Незначимі пропуски загалом ігноруються в інструкціях і запитах SQL, дозволяючи формувати код SQL з метою покращення читабельності.

Як і з багатьма стандартами, що мають місце в ІТ-індустрії, з мовою SQL виникла проблема, пов'язана з рішенням багатьох розробників ПЗ з використанням SQL про недостатню функціональність версії стандарту (що, в принципі, для ранніх версій SQL було певною мірою справедливо) і їхнім бажанням її розширити. Це призвело до того, що у різних виробників СУБД в застосуванні різні діалекти SQL, причому, несумісні між собою у більшості випадків.

До 1996 року питаннями відповідності комерційних реалізацій SQL стандарту займався переважно інститут NIST, який і встановлював рівень відповідності стандарту. Відтоді, як останній підрозділ, що займався СУБД, був розформований, всі зусилля з перевірки СУБД на відповідність стандарту були покладені на її виробника.

## Переваги

1. Незалежність від конкретної СУБД. Не зважаючи на наявність діалектів і відмінностей в синтаксисі, більшість текстів SQL-запитів, що містять, DDL і DML, можуть бути досить легко перенесені з однієї СУБД в іншу. Існують системи, розробники яких спочатку орієнтувалися на застосування щонайменше кількох СУБД (наприклад: система електронного документообігу може працювати як з Oracle, так і з Microsoft SQL Server та IBM DB2). Природно, що при застосуванні деяких специфічних для реалізації можливостей, такого рівня перенесення дуже важко досягти.

2. Наявність стандартів. Наявність стандартів і наборів тестів для виявлення сумісності та відповідності конкретній реалізації SQL загальноприйнятому стандарту тільки сприяє «стабілізації» мови. Щоправда, слід звернути увагу на той факт, що сам по собі стандарт місцями занадто формалізований і має з великі розміри, наприклад, Core-частина стандарту SQL:2003 містить понад 1300 сторінок тексту.

3. Декларативність. З а допомогою SQL програміст описує лише дані, які потрібно витягнути або модифікувати. Те, яким чином це зробити, вирішує СУБД безпосередньо при обробці SQL-запиту. Не слід вважати, що це повністю універсальний принцип — програміст описує набір даних для вибірки або модифікації, проте йому при цьому корисно уявляти, як СУБД інтерпретуватиме текст його запиту. Такі моменти стають особливо критичними при роботі з великими базами даних та зі складними запитами — чим складніше сконструйований запит, тим більше варіантів виконання він припускає. Ці варіанти можуть дуже відрізнятися за швидкістю виконання та використаними ресурсами, хоча результат (набір даних) має бути однаковим.

## Невідповідність реляційній моделі даних

Творець реляційної моделі даних Едгар Кодд, Крістофер Дейт та їхні прихильники вказують на те, що SQL не є істинно реляційною мовою. Зокрема, вони привертають увагу до таких проблем SQL:

- Рядки, що повторюються

- Невизначені значення (null)
- Явна вказівка порядку стовпчиків зліва направо
- Стовпчики без імені та імена стовпчиків, що повторюються
- Відсутність підтримки властивості «=»
- Використання вказівників
- Значна надлишковість

В опублікованому Крістофером Дейтом і Г'ю Дарвеном Третьому маніфесті<sup>[5]</sup>, вони декларували принципи СУБД наступного покоління та пропонували мову Tutorial D, яка є справді реляційною.

#### Складність

Хоча мову SQL було початково заплановано як засіб роботи кінцевого користувача, врешті-решт вона стала настільки складною, що перетворилася на інструмент програміста.

#### Відхилення від стандартів

Попри існування міжнародного стандарту ANSI (SQL-92, SQL:1999 та подальші версії), багато компаній, що розробляють СУБД (наприклад, Oracle, Sybase, Microsoft, MySQL), вносять свої зміни до мови SQL, що застосовується в розроблених ними СУБД. Цим вони створюють передумови відхилення від стандартів. Тож у кожній конкретній СУБД застосовуються специфічні діалекти мови SQL.

#### Складність роботи з ієрархічними структурами

Раніше SQL не пропонувала стандартного способу маніпуляції деревовидними структурами.

## 4 ОБЧИСЛЮВАЛЬНІ МЕТОДИ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ

Програма була розроблена з використанням C#, ArcGis, MS SQL. C# - це об'єктно-орієнтована мова програмування створена спеціально для роботи у середовищі Microsoft .NET Framework, розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research. Мова C# була розроблена з урахуванням переваг та недоліків інших мов, зокрема Java і C++.

Особливості мови C#:

- Портативність.
- Типи даних.
- Метапрограмування.
- Методи та функції.
- Синтаксис

Просторові дані мають в безлічі форматів, якими можна управляти в ArcCatalog і додавати шари в ArcMap. Бази геоданих – це формат зберігання просторових даних, розроблений спеціально для ARCGIS.

У ГІС просторові дані представляються із застосуванням двох моделей: векторної та растрової.

Векторна модель містить інформацію про точки, лінії, контури та поверхні, яка кодується і зберігається у вигляді набору координат X, Y. Місцеположення точки (точкового об'єкта), наприклад джерела емісії забруднювальних речовин, описується парою координат X, Y. Лінійні об'єкти, такі як річки, дороги або трубопроводи, зберігаються як набори координат X, Y. Полігональні об'єкти, такі як земельні й лісові ділянки, зберігаються у вигляді замкненого набору X, Y. Рельєфи, що є основою 3D-поверхневих карт, подаються наборами координат X, Y. Векторна модель зручна для опису дискретних об'єктів і неефективна для опису об'єктів із

неперервним характером зміни властивостей, таких як типи ґрунтів, види рослинності тощо.

Растрова модель є оптимальною для роботи з об'єктами, що мають безперервний характер зміни властивостей. Растрове зображення складається з окремих елементарних комірок, кожна з яких характеризується певним значенням. Цей спосіб представлення даних широко використовується для аерокосмічних знімків.

Таким чином, основним класом даних геоінформаційних систем (ГІС) є координатні дані, які містять геометричну інформацію й відображають просторовий аспект. Основні типи координатних даних: точка (вузли, вершини), лінія (незамкнута), контур (замкнута лінія), полігон (ареал, район). На практиці для побудови реальних об'єктів використовують більшу кількість даних (наприклад, висячий вузол, псевдовузол, нормальний вузол, покриття, шар та ін.).

Бази геоданих можуть бути або персональні, або розраховані на багато користувачів. Персональна база геоданих, як одна в цій книзі, зберігається у файлах баз даних Microsoft Access. Розраховані на багато користувачів бази геоданих (вони також називаються ArcSDE база геоданих) використовуються з реляційними системами управління базами даних (СУБД) таких, як Oracle, Informix, Microsoft SQL Server або DB2.

## **4.1 Середовище розробки**

Середовищем розробки було обрано Microsoft Visual Studio 2010.

У Visual Studio 2010 повністю перероблено інтерфейс з використанням Windows Presentation Foundation (WPF), запроваджено наступне покоління інструментів ASP.NET, є підтримка динамічних розширень в мовах програмування C# і Visual Basic, використовуються нові шаблони проектів, велика кількість нових бібліотек, що підтримують Windows 7.

В середовищі Microsoft Visual Studio 2010 з'явилася нова мова F#, Visual C++ підтримує стандарт C++0x. Для платформи .NET можна використовувати різні компілятори, незалежно від розробника. Розроблена велика кількість компіляторів, наприклад, Fortran, COBOL, Delphi, Lisp, Oberon, Perl, Python і Eiffel.

Середовище Visual Studio .NET підтримує декілька мов програмування вбудовані(C#, F#, J#, VB.NET, Jscript.NET, C++) та такі, що постачаються окремо, торонні — виробництва третіх фірм( ADA, APL, Boo, SCOBOL). Програма під час розробки має назву проектом. Він об'єднує всі ресурси (папки, файли, посилання) необхідні для створення програмного продукту.

Середовищем розробки Microsoft Visual Studio 2010 дозволяє створювати різні типи проектів:

1. web-додатки - додатки, доступ до яких виконується через браузер і які за запитом формують web-сторінку і відправляють її клієнтові;
2. консольний проект- виводить на вікно командного процесу;
3. windows-додатки використовують елементи інтерфейсу Windows, включаючи кнопки, форми, прапорці.

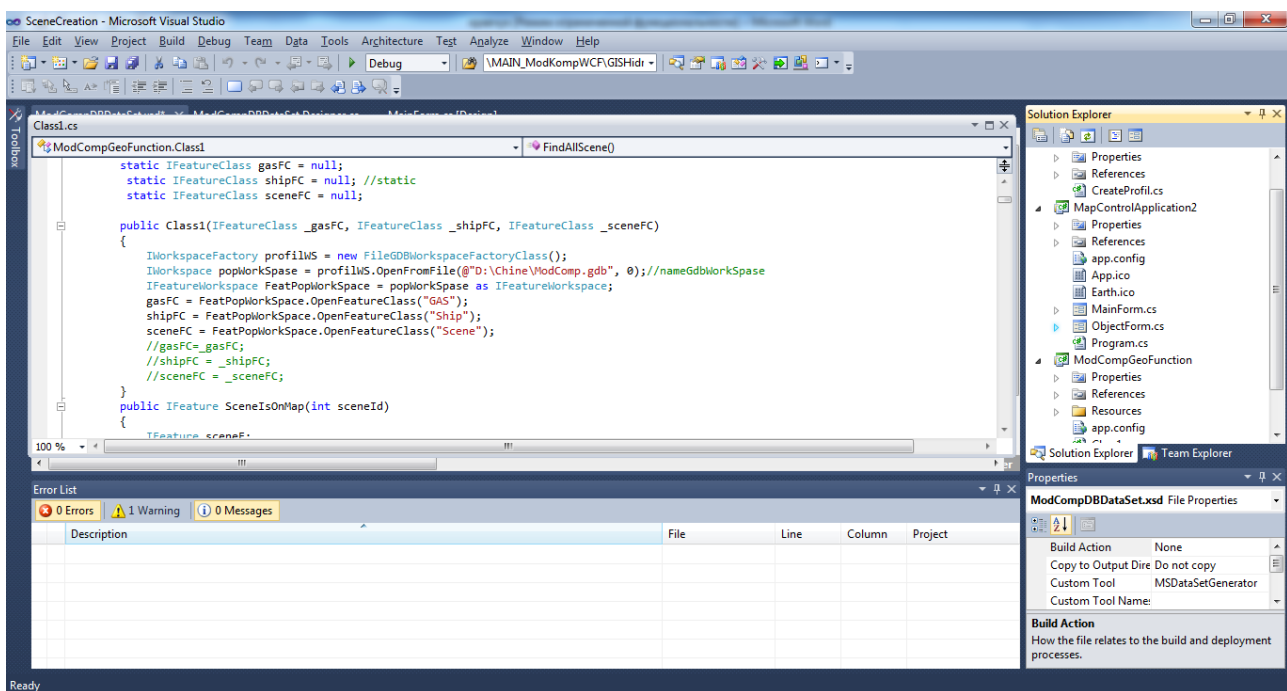


Рисунок 4.1 – Вигляд проекту в середовищі Visual Studio 2010

## 4.2 Технології, використані при розробці програмного продукту

При розробці проекту був використаний ArcObjects .NET SDK. ArcObjects є середовищем розробки сімейства геоінформаційних програмних продуктів американської компанії ESRI. Використовуючи C# для додатків, Visual Basic або Java SDK для ArcGIS, це дозволяє розробникам розширювати ці програми.

ArcObjects - це платформа розробки для таких модулів ArcGIS, як ArcMap, ArcCatalog і ArcScene. Програмні компоненти ArcObjects охоплюють повний діапазон функціональних можливостей, доступних в ArcInfo і ArcView для розробників програм.

ArcObjects - інфраструктура, яка дозволяє Вам створювати специфічні для даної предметної області компоненти з інших компонент. Компоненти ArcObjects взаємодіють, щоб обслужити кожен з функцій управління даними і функції уявлення карти, загальні для більшості ГІС додатків. ArcObjects забезпечує інфраструктуру для настройки додатків, яка дозволяє Вам сконцентруватися на обслуговуванні специфічних потреб ваших клієнтів. ArcObjects побудований з використанням технології Моделі компонентного об'єкту фірми Microsoft (COM - Component Object Model). Тому є можливість розширювати склад ArcObjects шляхом написання COM- компонент, використовуючи будь-який COM- сумісний мову розробки. Ви можете розширювати будь-яку частину архітектури ArcObjects точно так же, як це роблять розробники ESRI.

Щоб зрозуміти Модель компонентного об'єкту (всі COM-засновані технології), важливо зрозуміти, що це не об'єктно-орієнтована мова, протокол або стандарт. Модель компонентного об'єкту - більше ніж просто технологія; це - методологія розробки програм. Модель компонентного об'єкту визначає протокол, який з'єднує один програмний компонент або модуль з іншим. Використовуючи цей протокол, можна будувати програмні компоненти багаторазового використання, які можуть бути динамічно замінені в розподіленій системі. Модель компонентного об'єкту також визначає модель програмування, відому як інтерфейсні-засноване програмування. Компоненти полегшують програмне багаторазове використання,



тому що вони - окремі стандартні блоки, які можуть легко бути зібрані в великі системи. Модель компонентного об'єкту визначає об'єктну модель і вимоги програмування, які дають можливість об'єктам COM взаємодіяти з іншими об'єктами COM. Ці об'єкти можуть бути в межах одного процесу, в різних процесах, або навіть на віддалених машинах. Вони можуть бути написані на різних мовах і, можливо, розроблені дуже різними способами.

Для розробки в середовищі Microsoft Visual Studio 2010 було використано бібліотеки ERSI:

1. ESRI.ArcGIS.Geodatabase;
2. ESRI.ArcGIS.DataSourcesFile;
3. ESRI.ArcGIS.Geometry;
4. ESRI.ArcGIS.DataSourcesGDB.

ArcGIS.Geometry- Базовий клас для об'єктів геометрії. Цей клас не має конструктора. Бібліотека DataSourcesGDB містить реалізацію інтерфейсу програмування баз геоданих (API) для джерел даних бази даних. До таких джерел даних належать Microsoft Access, файлові бази геоданих, будь-яка система управління реляційними базами даних (СУБД), що підтримується ArcSDE, а також робочі області в пам'яті та нуля.

## 5. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Стуртура проекту поділена на модулі, які зв'язані між собою. Використання цих модулів дає можливість динамічно розширювати функціонал програми.

### 5.1 Опис реалізації

Структура проекту в Visual Studio складається з трьох модулів(рисунок 5.1):

- Головного модуля, який містить інтерфейс програми;
- Модуль створення профілю;
- Модуль формування сцени гідроакустичних експериментів.

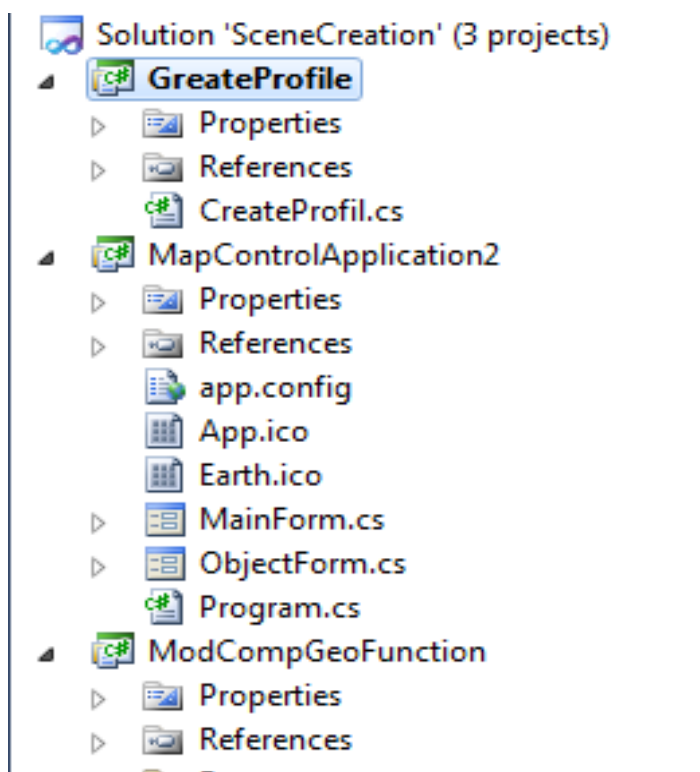


Рисунок 5.1 – Структура проекту

Для візуалізації карти був використаний ArcMap .NET Add in – компонент, який надає базовий функціонал: візуалізація карти, можливість додавання шарів та інше (рисунок 5.2).

При розробці програмного продукту був використаний принцип інверсії управління (inversion of control), також відомий як Dependency Injection.

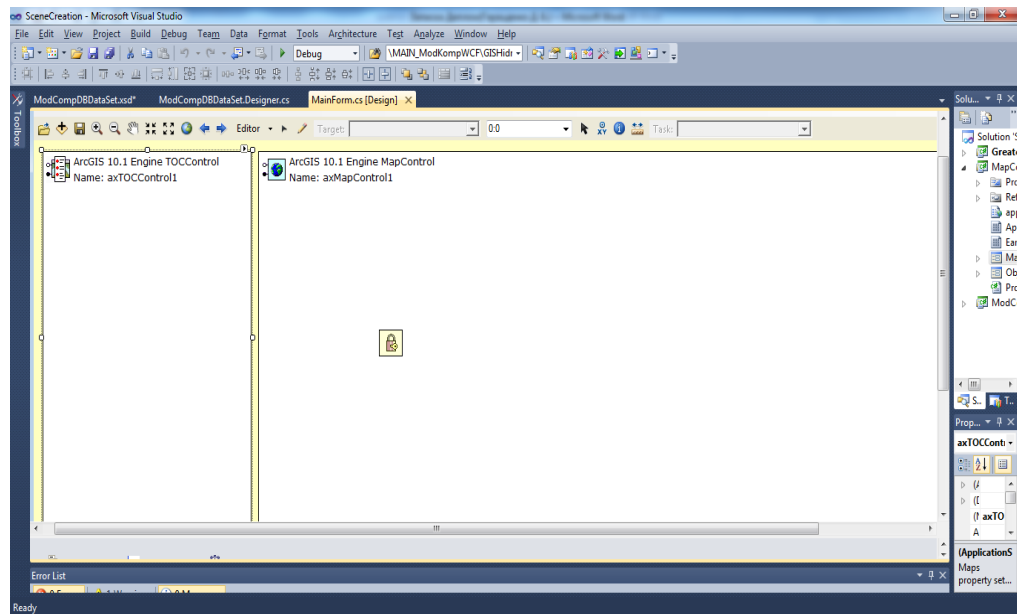


Рисунок 5.2 – Add in компонент ArcObject SDK

Графічний інтерфейс додатку складається з вікон та панелей управління, які можна налаштувати для повсякденних задач користувача без зміни програмного коду. Ці завдання виконуються за допомогою перетягування в додатку і не вимагають адміністративного втручання.

Додатки ArcGIS for Desktop за замовчуванням оснащені певним графічним призначеним для користувача інтерфейсом. Ви можете змінювати та розширювати функціонал графічного інтерфейсу користувача, а також надавати доступ до розроблених функцій для підвищення ефективності використання додатків.

Можна змінити деякі компоненти інтерфейсу:

- Розміщення панелі інструментів або вікон;
- Показ або приховування певних прикріплених вікон;
- Оптимальна для ваших завдань групування команд;

- Видалення непотрібних команд з панелей інструментів;
- Зміна значка або опису команди для її кращого впізнавання.

Ці завдання виконуються шляхом перетягування елементів інтерфейсу всередині програми. Всі настройки виконуються легко, не вимагають особливих прав доступу або навичок програмування. До того ж всі зміни в налаштуваннях автоматично зберігаються і відразу ж завантажуються в наступному сеансі роботи.

## 5.2 Опис Баз Даних

В Базі даних маємо вісім таблиць(рисунк 5.3)

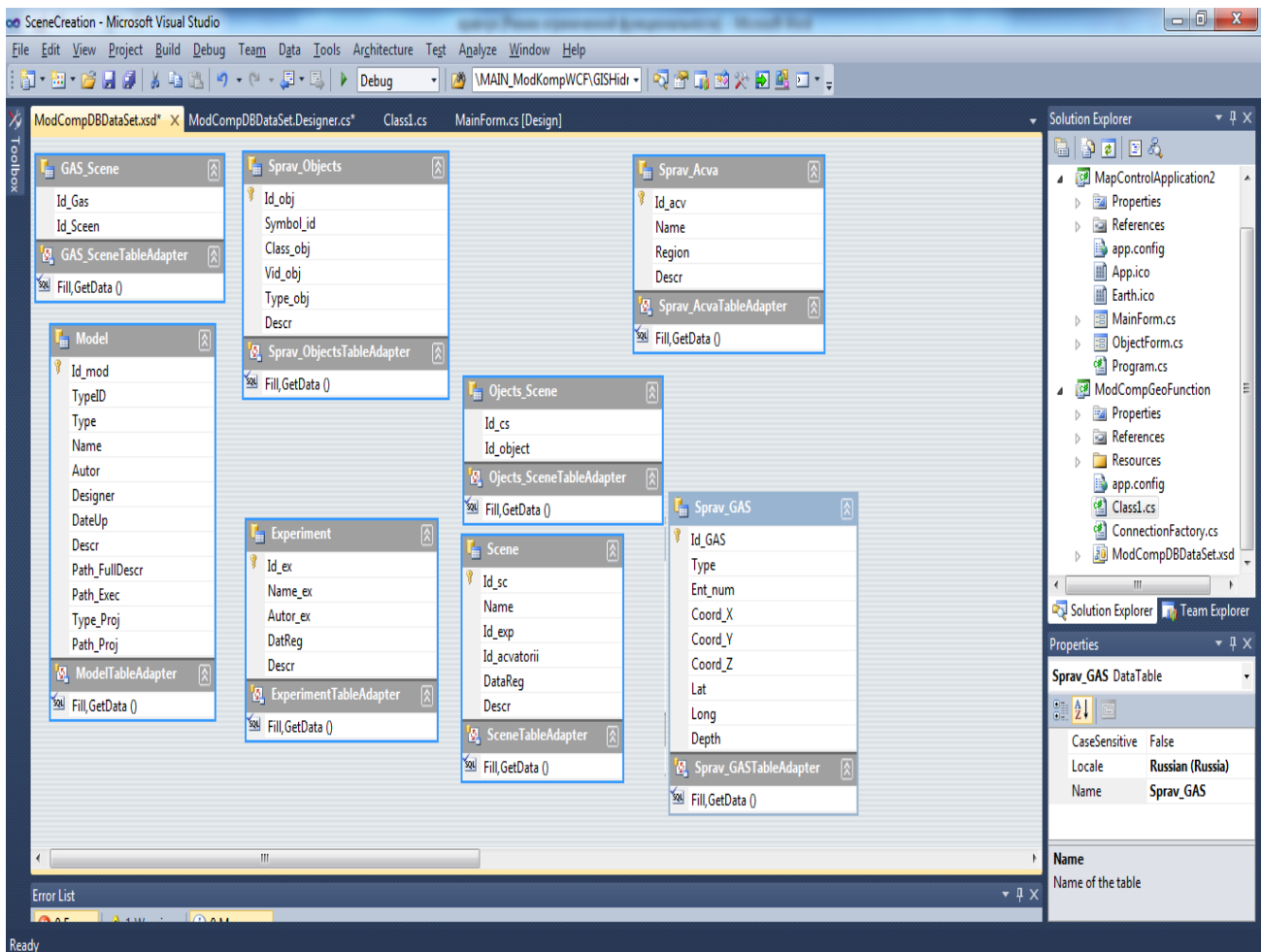


Рисунок 5.3 – Приклад Баз Даних

Для формування нової сцени гідроакустичних експериментів в складі моделюючого комплексу нам потрібно використати три таблиці(рис 5.4)

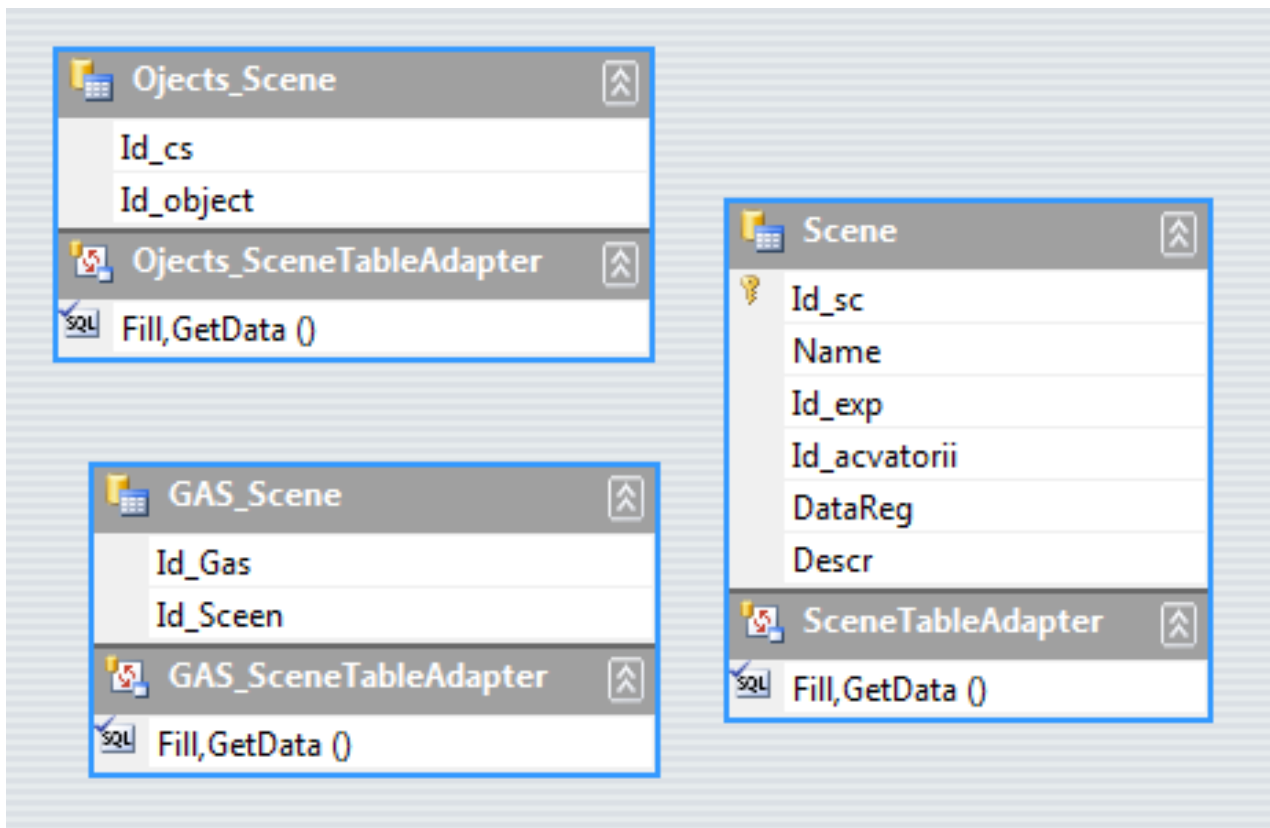


Рисунок 5.4— таблиці для створення нової сцени

При додаванні об'єктів на сцену моделювання, використовується база даних(рисунок 5.5), в яку заноситься координати їх знаходження.

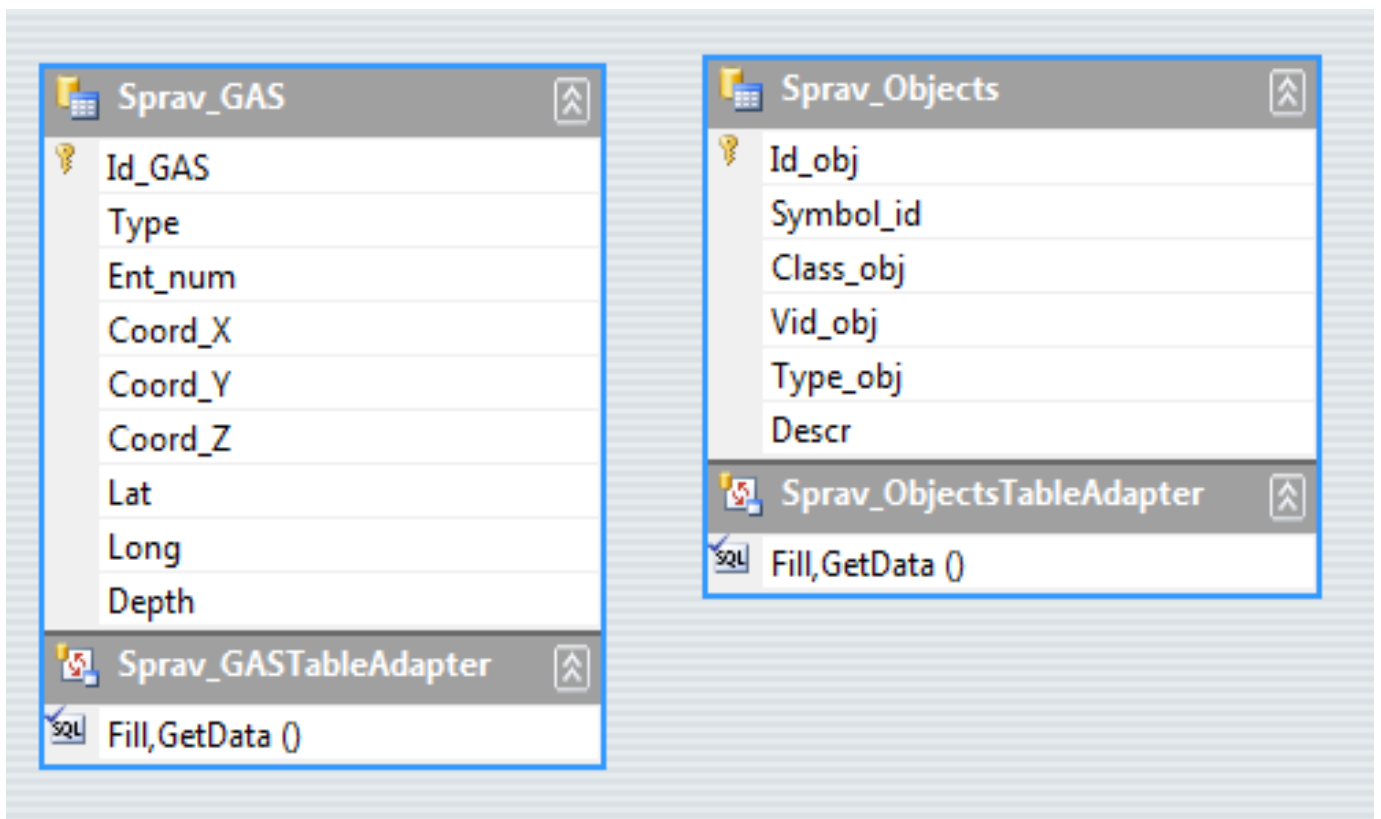


Рисунок 5.5— база даних об'єктів

### **5.3 Висновки до розділу**

Структура проекту в складається з трьох модулів (головного модуля, який містить інтерфейс програми, та двох інших). Графічний інтерфейс, який створений за допомогою ArcGis Desktop можна налаштувати відповідно до користувача.

## 6. РОБОТИ КОРИСТУВАЧА З ПРОГРАМОЮ

Даний розділ описує роботу користувача з програмним продуктом, який розроблений за допомогою середовища розробки Microsoft Visual Studio 2010.

### 6.1 Системні вимоги

Для встановлення розробленого програмного продукту потрібно мати встановлену операційну систему Windows 7 або більш пізнішу.

Для роботи з програмою мінімальна оперативна пам'ять — 512 Мб, рекомендується — 2GB.

### 6.2 Робота з програмою

Програма була розроблена мовою C# в середовищі розробки Microsoft Visual Studio 2010 з використанням функцій ArcObjects .NET SDK. Розроблена програма має графічний інтерфейс.

При запуску програми ми бачимо вікно управління експериментом (Рисунок— 6.1). На ньому зображені проведені експерименти зі збереженими даними.

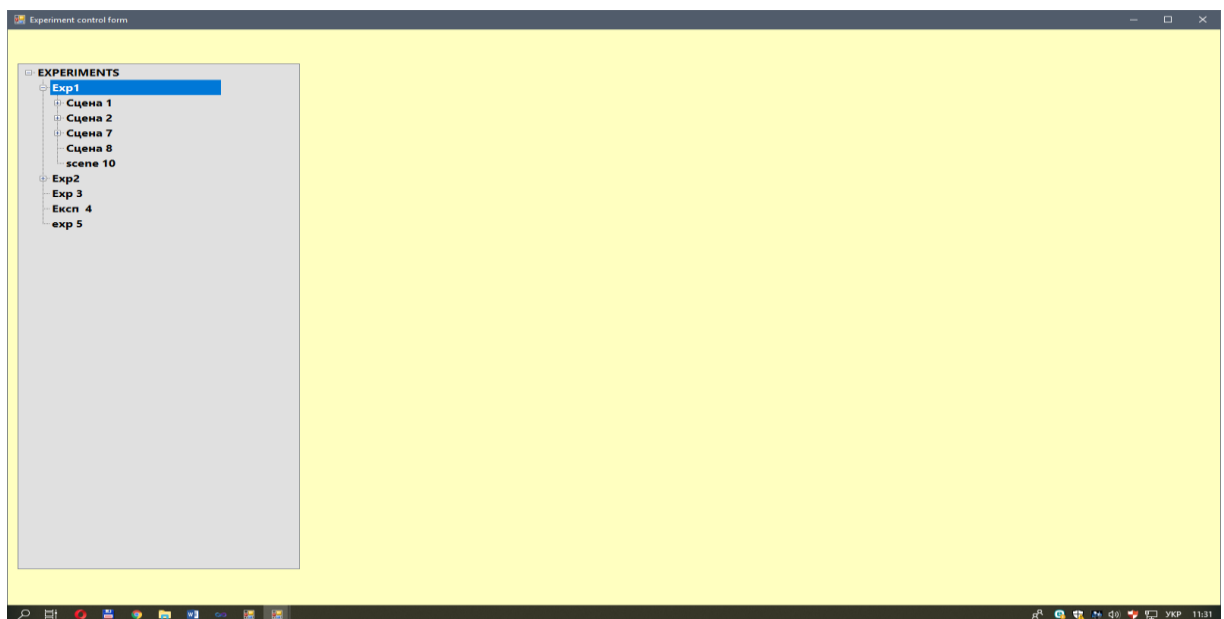


Рисунок 6.1— Вікно управління експериментом

Ми можемо вибрати будь-який експеримент і виконати його знову. Для цього потрібно вибрати експеримент та натиснути на нього, після чого нам відкриється нове вікно(рисунок 6.2) в якому будуть вже готові дані для формування сцени.

**Scene control form**

Show current scene (or build it for new )

**Review scenes table**

Id_sc	Id_exp	Id_acvatorii	DataReg	Descr	Name
1	1	1	10.10.2016	Первая сцена п...	Сцена 1
2	1	1	10.11.2016	Вторая сцена п...	Сцена 2
7	1	1	20.11.2017	Контрольная сц...	Сцена 7
1010	1	2	10.10.2016	Четвертая сце...	Сцена 8
1012	1	3	10.01.2019	scene 10 for exp 1	scene 10

Save all changes in database

**Scene account card**

Enter or change data about scenes for experiment

**Exp1**

**Scene name**

**Water area**

**Creation date**

**Description**

**Evaluation**

Fixed changes in Review table    Create new scene account card    Delete scene account card

Рисунок 6.2–Вибір сцени для експерименту

Для формування сцени на карті потрібно натиснути на кнопку Show current scene, після чого відкриється вікно уже з побудованими сценами(рисунок 6.3).



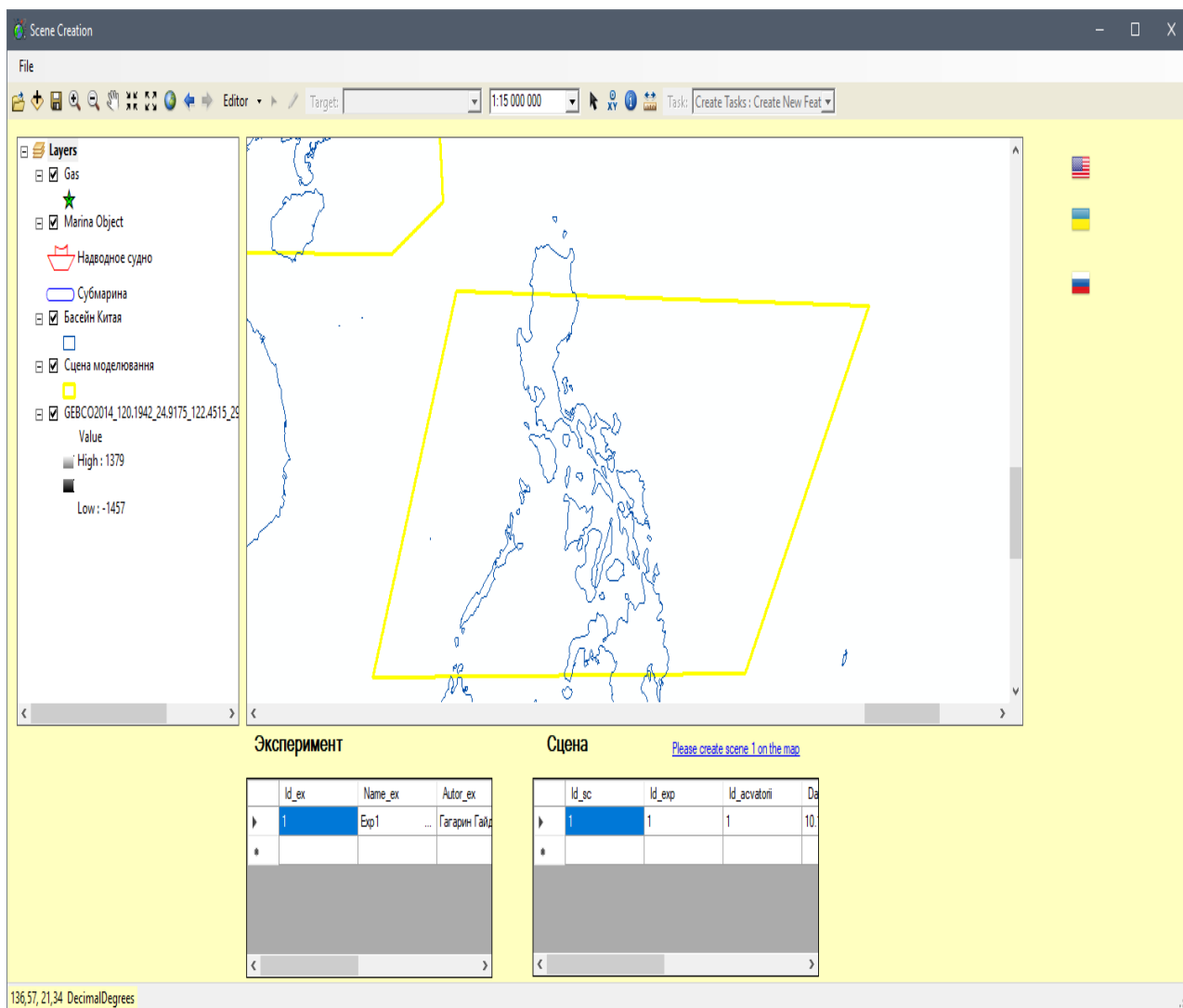


Рисунок 6.3– Візуалізація сцени

Також ми можемо створити нову сцену проведення гідроакустичного експерименту. Для цього нам потрібно натиснути меню Editor і в ньому вибрати Star Editing. Після цього можна вносити зміни на карті.

Сцени можна будувати будь-яких розмірів та форм. Побудуємо нову сцену експерименту на карті(рисунок 6.4).

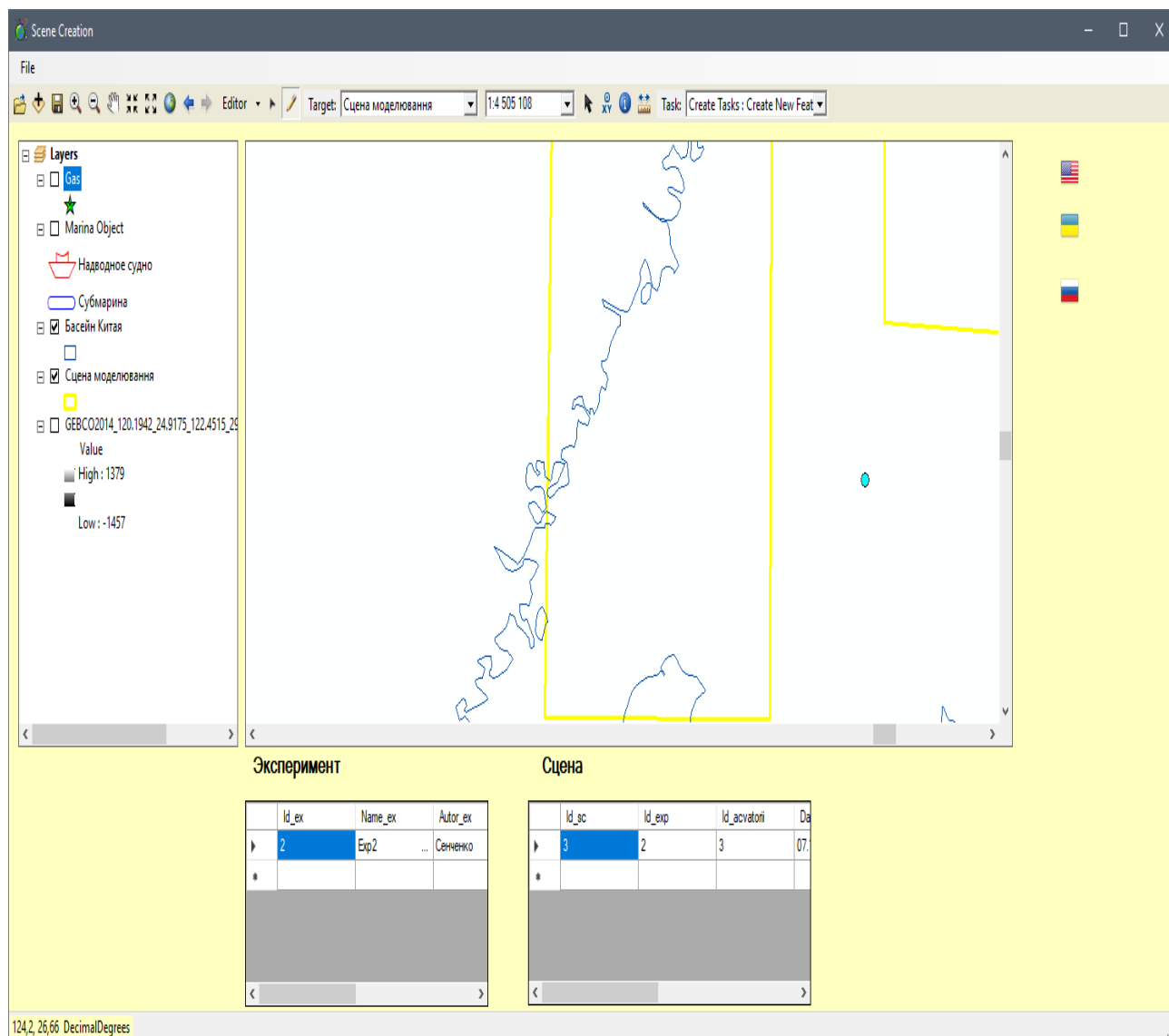


Рисунок 6.4—Створення нової сцени

Після формування нової сцени проведення гідроакустичного експерименту ми додаємо на карту батиметрію(рисунок 6.5) в зоні дослідження. Батиметрія – це рельєф морського дна, отриманий при виміру глибини спеціальним приладом.

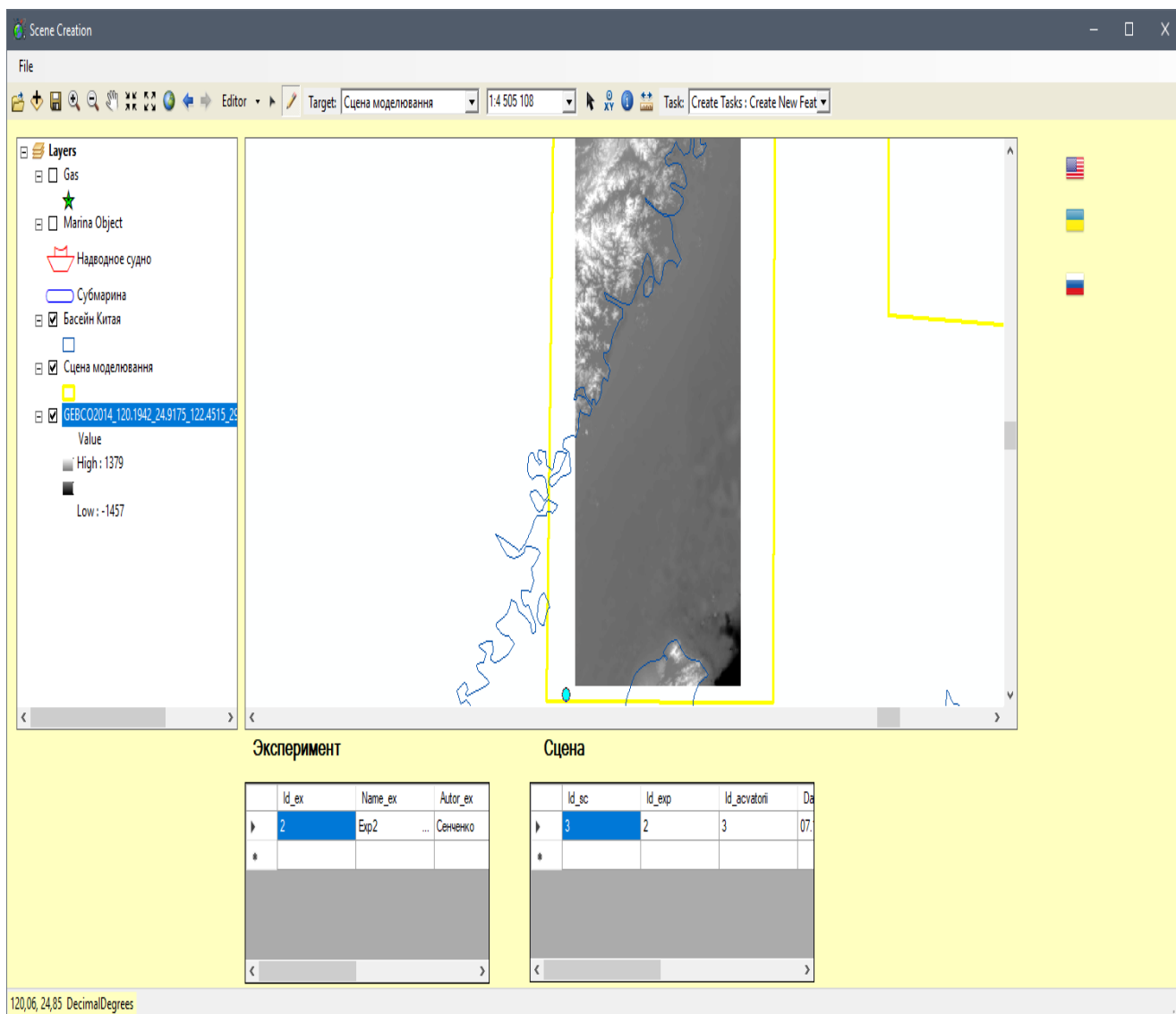


Рисунок 6.5—Додавання батиметрії

Після цього на сцену гідроакустичного експерименту потрібно додати судно, для чого в меню Target потрібно вибрати Marian Object, після цього відкриється вікно з базою даних, в якому потрібно вибрати тип судна (рисунк 6.6). Місце корабля можна змодельовати в будь-якому місці на сформованій сцені моделювання.

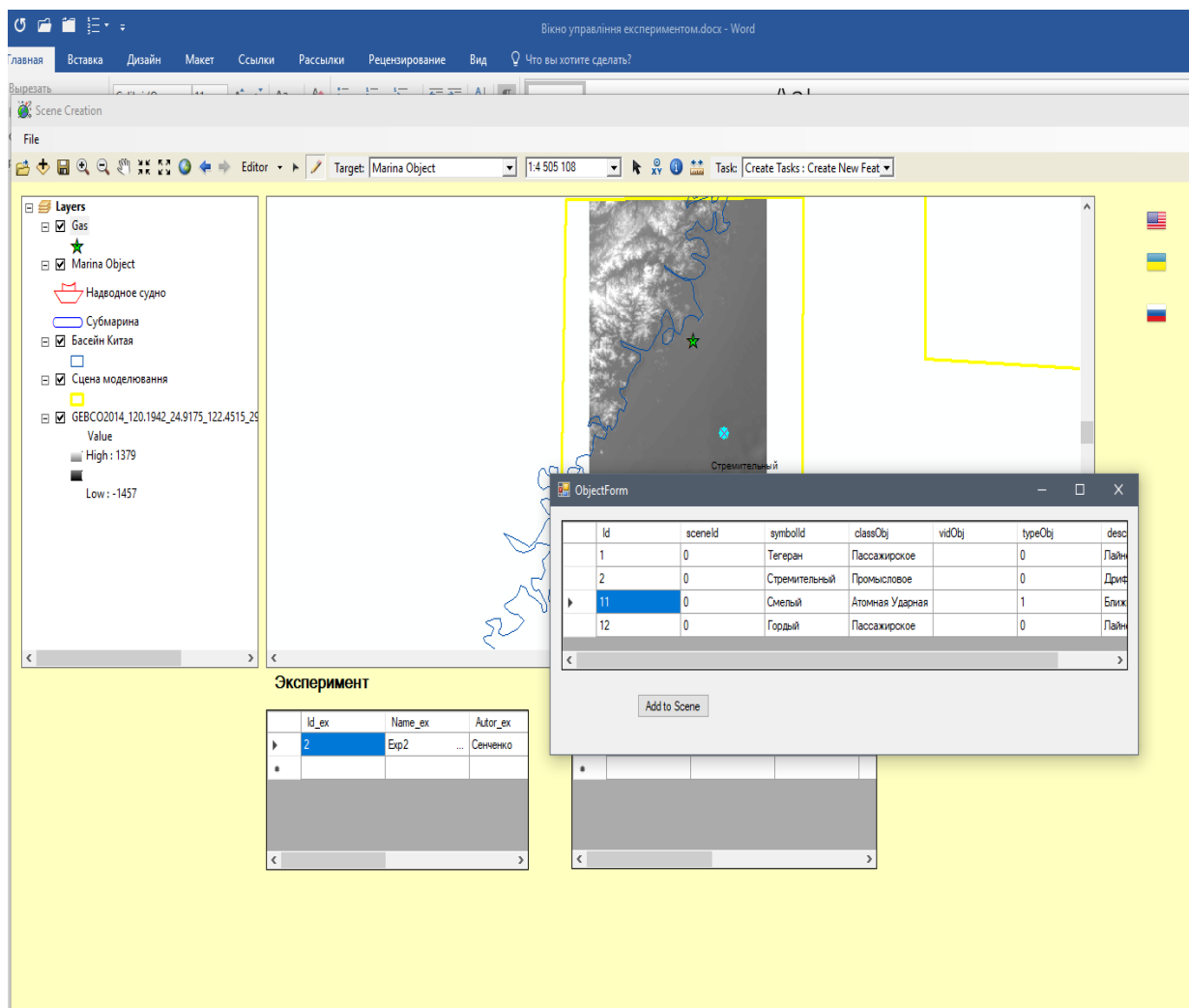


Рисунок 6.6– Додавання судна

Також потрібно додати гідроакустичний датчик. В меню Target потрібно вибрати GAS ,після цього відкриється вікно з базою даних,в якому потрібно вибрати ГАС(рисунок 6.7) . Місце гаса можна змоделювати в будь-якому місці на сформованій сцені моделювання.

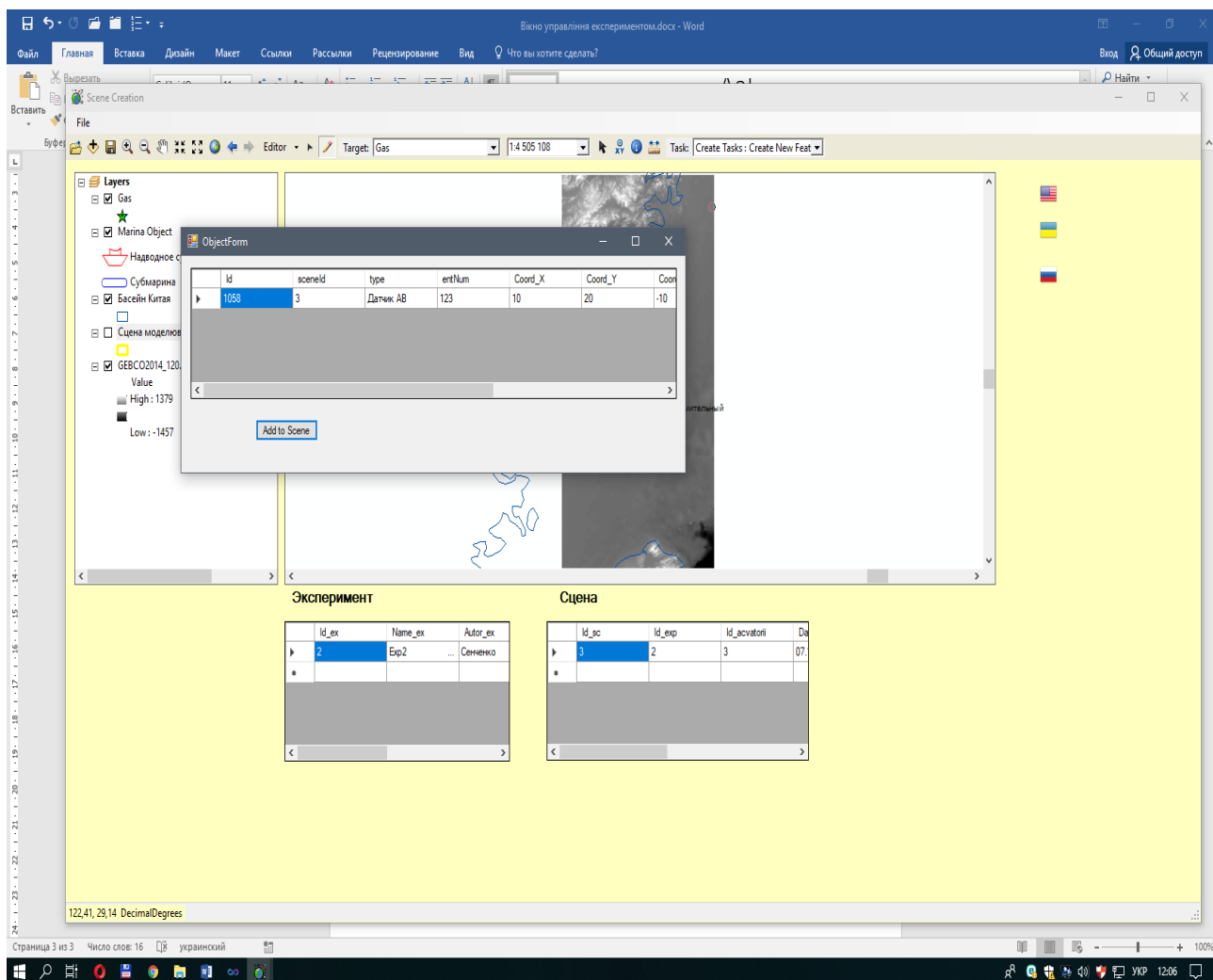


Рисунок 6.7– Додавання ГАСу

Після виконання всіх дій ми отримаємо загальний вигляд модулю формування сцени гідроакустичних експериментів в складі моделюючого комплексу(рисунок 6.8). Потрібно обов'язково зберегти зміни, всі зміни на карті зберігаються в Базі Даних (ModComDB).

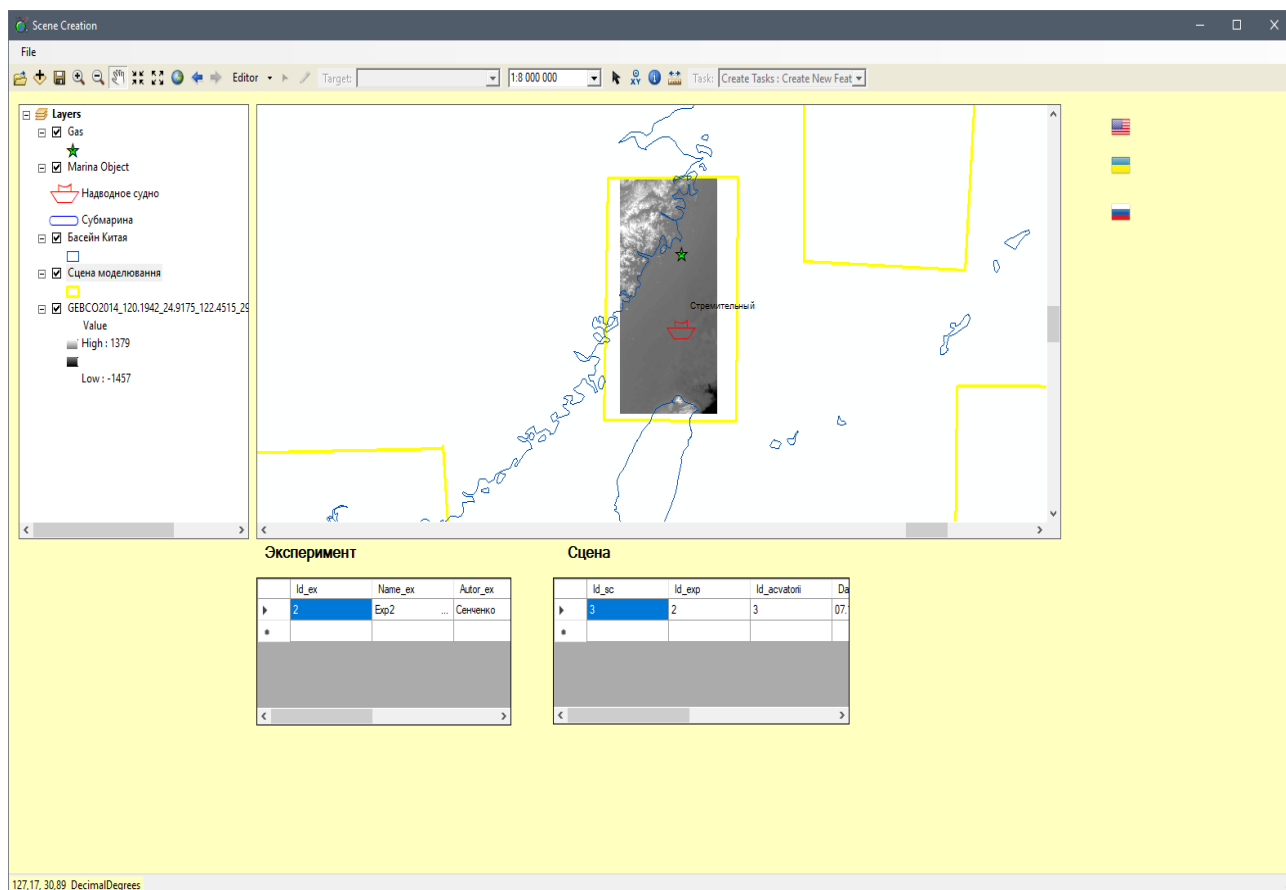


Рисунок 6.8 – Загальний вигляд модулю

Кожен із шарів на карті можна приховати в лівому вікні прибравши галочку, для того щоб повернути шар на місце потрібно повернути її на місце.

### 6.3 Висновки до розділу

Розроблений програмний продукт формує сцену проведення гідроакустичного експерименту в складі моделюючого комплексу з використанням ГІС. Інтерфейс зрозумілий та легкий для користування.

## **ВИСНОВКИ**

Під час виконання дипломної роботи був розроблений програмний продукт Для проведення гідроакустичних експериментів з використанням цифрових моделей рельєфу. Були розроблені і описані алгоритми роботи програми.

При виконанні роботи провели моделювання сцени гідроакустичних досліджень з використанням ГІС, розроблено базу даних та реалізовано збереження інформації в ній.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ArcGis Spatial Analyst [Електронний ресурс] // ESRI – Режим доступу до ресурсу: <https://pro.arcgis.com/ru/pro-app/tool-reference/spatial-analyst/an-overview-of-the-spatial-analyst-toolbox.htm>
2. ArcGis Spatial Analyst [Електронний ресурс] // ESRI – Режим доступу до ресурсу: <https://pro.arcgis.com/ru/pro-app/tool-reference/spatial-analyst/an-overview-of-the-spatial-analyst-toolbox.htm>
3. ArcObject SDK [Електронний ресурс] // ESRI – Режим доступу до ресурсу: [http://help.arcgis.com/en/sdk/10.0/arcobjects\\_net/componenthelp/index.htm](http://help.arcgis.com/en/sdk/10.0/arcobjects_net/componenthelp/index.htm)
4. [analyst/an-overview-of-the-spatial-analyst-toolbox.htm](#)
5. GIS Hydro'99 / Introduction to GIS Hydrology. — ESRI International User Conference (CD-R), 1999.
6. Meel, Ir.J. Spread Spectrum – Introduction and Application/ Ir.J. Meel . – Sirius Communication, 1999.
7. Popovich V.V., Potapichev S.N., Sorokin R.P., Pan kin A.V. Intelligent GIS for Monitoring Systems Devel.



## ДОДАТОК 1

Формування сцени проведення гідроакустичних експериментів в складі  
моделюючого комплексу з використанням ГІС

Специфікація

УКР.НТУУ «КПІ ім. Ігоря Сікорського»\_ТЕФ\_АПЕПС\_ТМ52

Аркушів 2

Київ – 2019

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ «КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕ ПС_TM52 19Б 81-1	Кравчук О. В_TM52.docx	Пояснювальна записка
Компоненти		
УКР.НТУУ «КПІ ім. І.Сікорського». TM5213_19Б 12-01	ConnectionFactory.cs	Модул серверної частини
УКР.НТУУ «КПІ ім. І.Сікорського». TM5213_19Б 12-02	CreateProfil.cs	Модуль створення профілю
УКР.НТУУ «КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕ ПС_TM52 19Б 12-3	MapControlApplication	Модуль моделювання сцени

## ДОДАТОК 2

Формування сцени проведення гідроакустичних експериментів в складі  
моделюючого комплексу з використанням ГІС

Текст програми

УКР.НТУУ«КПІ ім. Ігоря Сікорського»\_ТЕФ\_АПЕПС\_ТМ52\_19Б 12-1

Аркушів 6

Київ – 2019

## УКР.НТУУ«КПІ ім. Ігоря Сікорського»\_ТЕФ\_АПЕПС\_ТМ52\_19Б 12-1

```

namespace GreateProfile
{
    [Guid("64fab104-9291-4ccf-b542-bea64a1a78d6")]
    [ClassInterface(ClassInterfaceType.None)]
    [ProgId("GreateProfile.Class1")]
    public class CreateProfil
    {
        private IFeatureClass oceanProfils;
        IFeatureClass gasFC;
        IFeatureClass shipFC;
        IFeatureClass sceneFC;
        public List<OceanProperties> oceanProperties;
        public static int sceneIdFromDb;
        int modifiedValue;
        public CreateProfil(string nameProfilFC, string pathWS, string
fileNameOceanProperty, ObjectClassEventListener objectClassEventListeneShip,
ObjectClassEventListener objectClassEventListeneGAS)
        {
            IWorkspaceFactory profilWS = new FileGDBWorkspaceFactoryClass();
            IWorkspace popWorkSpase = profilWS.OpenFromFile(pathWS, 0);
            IFeatureWorkspace FeatPopWorkSpace = popWorkSpase as IFeatureWorkspace;
            oceanProfils = FeatPopWorkSpace.OpenFeatureClass(nameProfilFC);
            oceanProperties = GetOceanProp(fileNameOceanProperty);
            gasFC = FeatPopWorkSpace.OpenFeatureClass("Gas");
            // objectClassEventListeneGAS.init(gasFC as IObjectClass);
            IWorkspaceFactory sceneWSF = new FileGDBWorkspaceFactoryClass();
            IFeatureWorkspace sceneWS =
sceneWSF.OpenFromFile(@"D:\ArcData\Chine\TemplateData.gdb", 0) as IFeatureWorkspace;
            sceneFC = sceneWS.OpenFeatureClass("Scene");
            ObjectClassEventListenerForScene eventScene = new
ObjectClassEventListenerForScene(sceneFC as IObjectClass, 2, 1);
            shipFC = FeatPopWorkSpace.OpenFeatureClass("Ship");

            // objectClassEventListeneShip.init(shipFC);
            //IFeatureCursor gacCur = gasFC.Search(null, false);
            //gasPoint = gacCur.NextFeature().Shape as IPoint;
            //IFeatureCursor profilCur = profileFC.Update(null, false);
            //profilFeature = profilCur.NextFeature(); ;

        }

        public CreateProfil(string nameProfilFC, string pathWS, string
fileNameOceanProperty)
        {
            IWorkspaceFactory profilWS = new FileGDBWorkspaceFactoryClass();
            IWorkspace popWorkSpase = profilWS.OpenFromFile(@"D:\Chine\ModComp.gdb",
0);
            IFeatureWorkspace FeatPopWorkSpace = popWorkSpase as IFeatureWorkspace;
            oceanProfils = FeatPopWorkSpace.OpenFeatureClass(nameProfilFC);
            oceanProperties = GetOceanProp(fileNameOceanProperty);
            gasFC = FeatPopWorkSpace.OpenFeatureClass("Gas");
            IWorkspaceFactory sceneWSF = new FileGDBWorkspaceFactoryClass();
            IFeatureWorkspace sceneWS = sceneWSF.OpenFromFile(@"D:\Chine\ModComp.gdb", 0) as
IFeatureWorkspace;
            sceneFC = sceneWS.OpenFeatureClass("Scene");
            ObjectClassEventListenerForScene eventScene = new
ObjectClassEventListenerForScene(sceneFC as IObjectClass, 2, 1);

```

```

        shipFC = FeatPopWorkSpace.OpenFeatureClass("Ship");

        //IFeatureCursor gacCur = gasFC.Search(null, false);
        //gasPoint = gacCur.NextFeature().Shape as IPoint;
        //IFeatureCursor profilCur = profileFC.Update(null, false);
        //profilFeature = profilCur.NextFeature(); ;
    }
    public IFeature SceneIsOnMap(int sceneId)
    {
        IFeature sceneF;
        IQueryFilter sceneQF = new QueryFilterClass();
        sceneQF.WhereClause = "SceneId="+sceneId.ToString();
        IFeatureCursor scecenFCur = sceneFC.Search(sceneQF, false);
        if((sceneF=scecenFCur.NextFeature())==null) return null;
        return sceneF;
    }
    public static List<OceanProperties> GetOceanProp(string fileName)
    {
        //string connectionString = "";
        string r_string = "";
        string[] split = null;
        List<OceanProperties> _oceanProps = new List<OceanProperties>();
        System.IO.StreamReader tr = System.IO.File.OpenText(fileName);
        do
        {
            r_string = tr.ReadLine();
            if (r_string == null) break;
            split = r_string.Split(';');
            OceanProperties oceanProp = new OceanProperties(Convert.ToInt32(split[2]),
            Convert.ToInt32(split[1]), Convert.ToDouble(split[3]), Convert.ToDouble(split[4]),
            Convert.ToDouble(split[5]), Convert.ToDouble(split[6]));
            _oceanProps.Add(oceanProp);
        } while (r_string.Length != 0);

        tr.Close();
        return _oceanProps;
        throw new NotImplementedException();
    }
    /// <summary>
    /// Определение линии профиля
    /// </summary>
    /// <param name="xGas"> Координата X датчика </param>
    /// <param name="yGas"> Координата Y датчика</param>
    /// <param name="L" > Длина профиля</param>
    /// <param name="fi"> Азимут профиля</param>
    /// <returns></returns>
    public IFeature GetOceanProfil(double xGas, double yGas, double L, double fi)
    {
        IFeatureCursor profilCur = oceanProfils.Update(null, false);
        IFeature oceanProfil = profilCur.NextFeature();
        IPoint gasPoint = new PointClass();
        gasPoint.X = xGas;
        gasPoint.Y = yGas;
        IPoint _p = new PointClass();
        double X2 = xGas + (L / 111000) * Math.Cos((fi / 180) * Math.PI);
        double Y2 = yGas + (L / 111000) * Math.Sin((fi / 180) * Math.PI);
        _p.X = X2;
        _p.Y = Y2;
        if (gasPoint != null)
        {
            ICurve profPL = new PolylineClass();

```

```

        profPL.FromPoint = gasPoint;
        profPL.ToPoint = _p;
        oceanProfil.Shape = profPL as IGeometry;
        oceanProfil.Store();
    }
    return oceanProfil;
}
public List<OceanLineProfil> CreateSlopeProfile(IFeature lineProfile, IRaster2
batymetry, ref List<PointOceanProfil> resultMedMesures, string oceanProperties)
{
    IPolyline routeLine = lineProfile.Shape as IPolyline;
    ISpatialReference _rasterSR = batymetry.GeodataXform.SpatialReference;
    resultMedMesures.Clear();
    List<OceanLineProfil> oceanLineProfils = new List<OceanLineProfil>(); ;
    double distance = .0;
    ISegmentCollection pSegmentColl = routeLine as ISegmentCollection;
    resultMedMesures.Clear();
    double sumAll = .0;
    double katet = 0.0075 * 110000;
    int colPrev = 0;
    int rowPrev = 0;
    double gammaPrev = 0;
    double delta = 10;
    OceanLineProfil oceanLineProfil = null;
    for (int j = 0; j < pSegmentColl.SegmentCount; j++)
    {
        double _m = 0;
        ILine _rls = pSegmentColl.get_Segment(j) as ILine;

        for (int i = 0; i < _rls.Length / 0.0075; i++)
        {
            IPoint outPoint = new PointClass();
            _rls.QueryPoint(0, 0.075 * i, true, outPoint);
            outPoint.Project(_rasterSR);
            double coX = outPoint.X;
            double coY = outPoint.Y;
            int col = batymetry.ToPixelColumn(coX);
            int row = batymetry.ToPixelRow(coY);
            // Визначити значення у координаті
            double _med = Convert.ToDouble(batymetry.GetPixelValue(0, col, row));
            _m = _m + 0.0075;
            distance = distance + 0.0075;
            double _step = 10.0;
            PointOceanProfil _pointProfil = new PointOceanProfil(coX, coY, _med,
Math.Abs(_med % 10), _step, lineProfile, oceanProperties);
            resultMedMesures.Add(_pointProfil);
            if (j == 0 && i == 0)
                //пропускаем вычисление угла
            {
                colPrev = col;
                rowPrev = row;
            }
            else
            {
                //вычисляем уклон
                double pixValue1 = Convert.ToDouble(batymetry.GetPixelValue(0,
colPrev, rowPrev));
                double pixValue2 = Convert.ToDouble(batymetry.GetPixelValue(0, col,
row));
                double gamma = (Math.Atan((pixValue2 - pixValue1) / katet) /
Math.PI) * 180;
                if (j == 0 && i == 1) //первый отрезок

```

```

        {
            oceanLineProfil = new OceanLineProfil();
            oceanLineProfil.gamma = gamma;
            oceanLineProfil.lineKoord = katet;
            oceanLineProfil.soundSpeed = _pointProfil.SoundSpeeds[0];
            oceanLineProfil.h = _pointProfil.H;
            gammaPrev = gamma;
        }
        else
        {
            if ((Math.Abs(gamma - gammaPrev) / Math.PI) * 180 < delta)
            {
                oceanLineProfil.lineKoord = oceanLineProfil.lineKoord +
katet;

            }
            else
            {
                double lineKoord = oceanLineProfil.lineKoord;
                oceanLineProfils.Add(oceanLineProfil);
                oceanLineProfil = new OceanLineProfil();
                oceanLineProfil.lineKoord = lineKoord + katet;
                oceanLineProfil.gamma = gamma;
                oceanLineProfil.teta = gamma;
                oceanLineProfil.soundSpeed = _pointProfil.SoundSpeeds[0];
                oceanLineProfil.h = _pointProfil.H;
            }
        }
    }

    }
    return oceanLineProfils;
}
}

```

```

public class PointOceanProfil
{
    public double X { get; set; }
    public double Y { get; set; }
    public double H { get; set; }
    public double GroundType { get; set; }
    public List<double> SoundSpeeds;
    private double coX;
    private double coY;
    private double _med;
    private double p;
    private int p_2;
    private IFeature profilFeature;
    private double _step;
    // private List<OceanProperties> oceanProps;

    public double StepSoundSpeed { get; set; }
    public PointOceanProfil(double _X, double _Y, double _H, double _GroundType, double
_stepSoundSpeed, IFeature featProfil, string _fileName)
    {
        X = _X;
        Y = _Y;
    }
}

```

```

        H = _H;
        GroundType = _GroundType;
        List<OceanProperties> _ocProps = CreateProfil.GetOceanProp(_fileName);
        OceanProperties[] pointOceanProperties = _ocProps.Where(po => po.z1 >= _H &&
po.season == 1 && po.watertype == 1).OrderBy(po => po.z1).ToArray();
        SoundSpeeds = new List<double>();
        for (int i = 0; i < pointOceanProperties.Length; i++)
        {
            double _h = 0;
            OceanProperties op = pointOceanProperties[i];
            if (i == 0)
            {
                _h = op.z2; //op.z1 - ((op.z1 - op.z2) % _stepSoundSpeed) *
_stepSoundSpeed;
            }
            while (_h < op.z1)
            {
                SoundSpeeds.Add(soundSpeeds(op.temp, op.solt));
                _h = _h + _stepSoundSpeed;
            }
        }
    }

    //public PointOceanProfil(double coX, double coY, double _med, double p, double
_step, IFeature profilFeature, List<global::MapControlApplication2.OceanProperties>
oceanProps)
    //{
    //    // TODO: Complete member initialization
    //    this.coX = coX;
    //    this.coY = coY;
    //    this._med = _med;
    //    this.p = p;
    //    this._step = _step;
    //    this.profilFeature = profilFeature;
    //    this.oceanProps = oceanProps;
    //}

    private double soundSpeeds(double p, double p_2)
    {
        return 100 * p * p_2;
        throw new NotImplementedException();
    }
}

public class OceanProperties
{
    public int season { get; set; } //c
    public int watertype { get; set; }
    public double z1 { get; set; }
    public double z2 { get; set; }
    public double temp { get; set; }
    public double solt { get; set; }
    public OceanProperties(int _season, int _watertype, double _z1, double _z2, double
_temp, double _solt)
    {
        season = _season;
        watertype = _watertype;
        z1 = _z1 * (-1);
        z2 = _z2 * (-1);
    }
}

```



```

        temp = _temp;
        solt = _solt;
    }
}
public class OceanLineProfil
{
    public double lineKoord { get; set; }
    public double h { get; set; }
    public double teta { get; set; }
    public double gamma { get; set; }
    public double temp { get; set; }
    public double solt { get; set; }
    public double soundSpeed { get; set; }
}
public class ObjectClassEventListenerForScene
{
    // Member variables for field index storage.
    private int createdFieldIndex = -1;
    private int modifiedFieldIndex = -1;

    public ObjectClassEventListenerForScene(IObjectClass objectClass, int
        createdFieldIndex, int modifiedFieldIndex)
    {
        // Cast the object class to the event interface.
        IObjectClassEvents_Event objectClassEvent = (IObjectClassEvents_Event)
            objectClass;

        // Instantiate the delegate types and add them to the events.
        objectClassEvent.OnCreate += new IObjectClassEvents_OnCreateEventHandler
            (OnCreate);
        objectClassEvent.OnChange += new IObjectClassEvents_OnChangeEventHandler
            (OnChange);

        // Store the index parameters in member variables.
        this.createdFieldIndex = createdFieldIndex;
        this.modifiedFieldIndex = modifiedFieldIndex;
    }

    public void OnCreate(IObject createdObject)
    {
        if (createdObject.Class.AliasName == "Scene")
        {
            // Set the creation date to Now. Note that Store() is not needed.
            createdObject.set_Value(createdFieldIndex, CreateProfil.sceneIdFromDb);
        }
    }

    public void OnChange(IObject changedObject)
    {
        // Set the modification date to Now. Note that Store() is not needed.
        changedObject.set_Value(modifiedFieldIndex, DateTime.Now);
    }
}
}

```

## ДОДАТОК 3

Формування сцени проведення гідроакустичних експериментів в складі  
моделюючого комплексу з використанням ГІС

Опис програми

УКР.НТУУ «КПІ ім. Ігоря Сікорського»\_ТЕФ\_АПЕПС\_ТМ52\_19Б 13-2

Аркушів 8

Київ – 2019

## АНОТАЦІЯ

Додаток надає можливість моделювати та генерувати гідроакустичні сигнали променевим методом.

Розроблене програмне забезпечення дозволяє моделювати сцену гідроакустичних експериментів в складі моделюючого комплексу.

Програмний продукт створений мовою програмування C# на платформі Microsoft .Net з використанням фреймворків ADO.Net, Entity Framework. Розробка інтерфейсу додатку відбувалася за допомогою API WinForms та ArcObject SDK. В якості середовища розробки – Microsoft Visual Studio 2010.

# ЗМІСТ

1. Загальні відомості .....	4
2. Функціональне призначення .....	5
3. Опис логічної структури.....	6
4. Використовувані технічні засоби .....	7
5. Вхідні і вихідні дані .....	8

## **ЗАГАЛЬНІ ВІДОМОСТІ**

Відповідно до теми дипломної роботи, програма має назву «Формування сцени».

Програма працює в вікні не потребує доступу інтернет і не потребує встановлення на ПК зайвого ПО, що забезпечує практичність та швидкість.

Система була написана мовою C#.

## **ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ**

Розроблений програмний засіб покликаний вирішити задачу моделювання сцени гідроакустичних досліджень. Це було реалізовано за допомогою кількох функцій системи, а саме:

- моделювання гідроакустичного сигналу за допомогою розробленого алгоритму;
- завантаження згенерованого файлу;
- відображення графічної візуалізації гідроакустичного сигналу

## ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Програмний продукт складається з клієнтської та серверної частини.

Загальний принцип роботи додатку такий:

- 1) користувач вводить початкові дані;
- 2) Натискає на кнопку “Моделировать”
- 3) метод, що викликається в обробнику виконує запит;
- 4) результати виконання методу виводяться на сторінку.

В першу чергу завантажується форма з усіма її компонентами: кнопки, текстові поля, таблиці та акваторія. Користувач вводить всі необхідні дані та додає направлення на акваторії.

Після натискання на кнопку “Моделировать”, що розміщена під формою, викликається метод `modelling`, в який передаються всі початкові параметри. Він виконує запит до сервера та передає всі ці дані. Сервер в свою чергу починає проведення розрахунків за написаним алгоритмом. Після успішного завершення він повертає результат на клієнтську сторону у вигляді масиву з порахованими показниками сигналу для його візуалізації та силку на скачування згенерованого файлу з `dat` розширенням.

Після закінчення роботи з цими даними, користувач без перезавантаження сторінки може очистити всі введені початкові дані і направлення відповідними кнопками та почати генерацію наступного сигналу.



## **ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ**

Для організації доступу до програмного продукту потрібно мати комп'ютер або ноутбук.

Кінцевим користувачам для роботи з програмою не потрібно встановлювати зайві ПО.

## **ВХІДНІ І ВИХІДНІ ДАНІ**

Вхідними даними є:

- глибина водного середовища;
- початковий час;
- кінцевий час;
- глибина гідрофона;
- частота коливань;
- амплітуда коливань;
- направлення та відомості про них.

Вихідними даними є:

- збережені дані про місце знаходження корабля та гідроакустичного датчика.